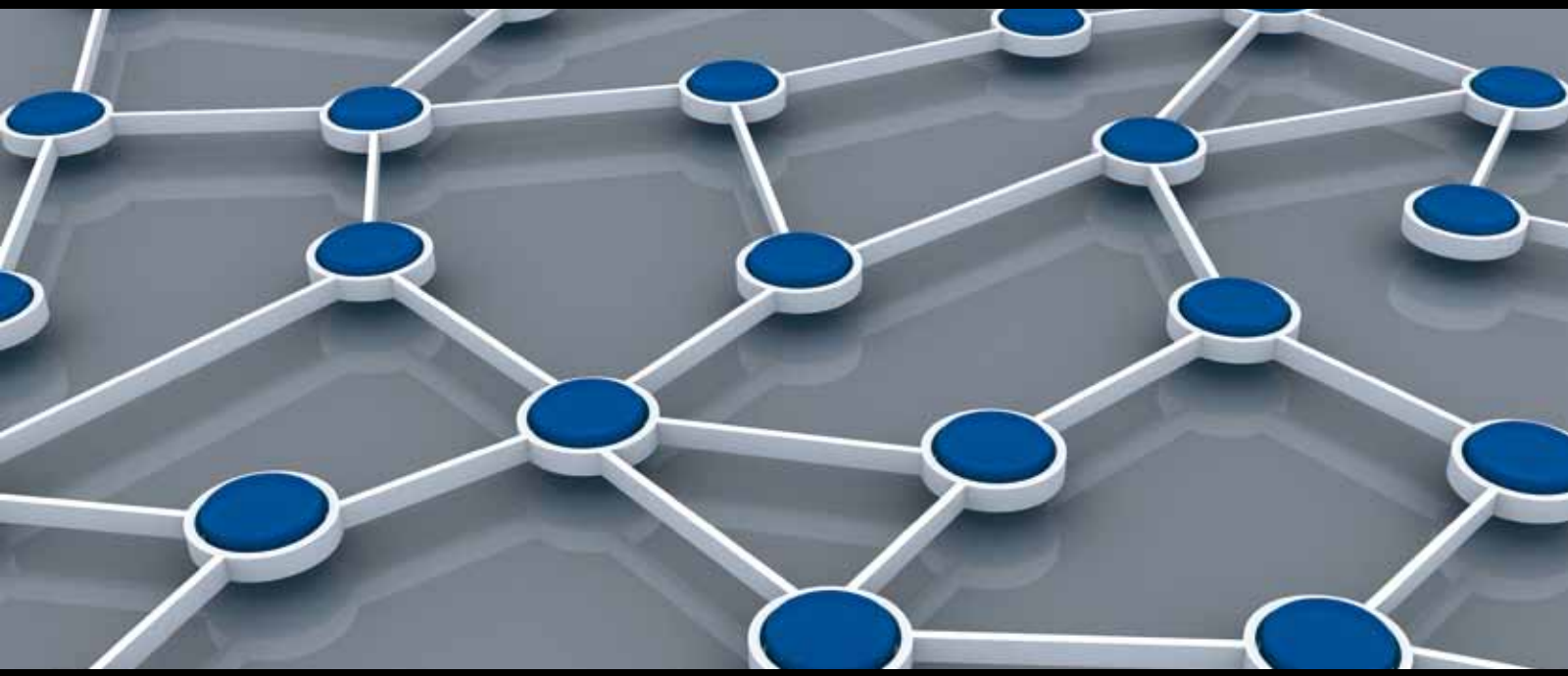
International Journal of Distributed Sensor Networks

Trust, Security, and Privacy in Next-Generation Wireless Sensor Networks

Guest Editors: Muhammad Khurram Khan, Yang Xiang, Shi-Jinn Horng, and Hsiao-Hwa Chen



Trust, Security, and Privacy in Next-Generation Wireless Sensor Networks

International Journal of Distributed Sensor Networks

Trust, Security, and Privacy in Next-Generation Wireless Sensor Networks

Guest Editors: Muhammad Khurram Khan, Yang Xiang, Shi-Jinn Horng, and Hsiao-Hwa Chen

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 956736, 2 pages
http://dx.doi.org/10.1155/2013/956736

Editorial

Trust, Security, and Privacy in Next-Generation Wireless Sensor Networks

Muhammad Khurram Khan,[1] Yang Xiang,[2] Shi-Jinn Horng,[3] and Hsiao-Hwa Chen[4]

[1] *Center of Excellence in Information Assurance, King Saud University, P.O. Box 92144, Riyadh 11653, Saudi Arabia*
[2] *Network Security and Computing Lab, School of Information Technology, Deakin University, 221 Burwood Highway, Burwood, VIC 3125, Australia*
[3] *Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Section 4, 43 Kee-Lung Road, Taipei 106, Taiwan*
[4] *Department of Engineering Science, National Cheng Kung University, Tainan 70101, Taiwan*

Correspondence should be addressed to Muhammad Khurram Khan; mkhurram@ksu.edu.sa

Received 9 September 2013; Accepted 9 September 2013

Recent advances in wireless communication technologies and applications have enabled the large-scale deployment of wireless sensor networks (WSNs). These networks have applications in several important and critical areas, such as military, critical infrastructure monitoring, healthcare, environment monitoring, and manufacturing. Security and privacy are critical issues to many sensor network applications, and it is a paramount concern for the growth of WSNs. Providing security and privacy to WSNs is challenging due to the open nature of wireless communication and the limited capabilities of sensor nodes in terms of processing power, storage, bandwidth, and energy. Moreover, widespread and unrestricted deployment of WSNs makes them exposed to a number of security and privacy vulnerabilities.

This special issue is aimed at fostering the latest development in the design, implementation, and evaluation in the fields of trust, security, and privacy of WSNs. Original contributions that provide novel theories, frameworks, and solutions to challenging problems of trust, security, and privacy were solicited for this special issue. As a whole, this special issue contains a diverse collection of 17 high-quality papers authored by eminent academicians and researchers in the field.

The paper by S. Irum et al. presents a hybrid key agreement scheme for wireless body area networks (WBANs) for both intra-WBAN and inter-WBAN communications. The authors compare their proposed scheme with the existing work and show through simulations and theoretical analysis that the proposed scheme is a better choice for WBANs in terms of security, storage efficiency, and energy consumption.

The paper by M. Faisal et al. integrates a lightweight PKI that utilizes ECC and AES cryptoalgorithms with a secure data transmission service via multipath in an efficient manner, in terms of resource and power consumption. The proposed protocols enhance the security of WSN by providing mutual authentication between neighbor nodes and sending the data via multipath for redundancy.

The paper by Y. Zhang et al. presents a secret sharing-based key management (SSKM) for wireless sensor network. SSKM utilizes the advantages of hierarchical architecture and adopt two-level key management and authentication mechanism. Different from previous works, the SSKM distributes keys based on secret sharing mechanism by the clustered architecture, which not only localizes the key things but also keeps scalability.

The paper by A. Ye et al. proposes an improved positioning algorithm for RFID tagged objects. They introduce a nonlinear interpolation method and the concept of credibility of reference tags, which not only achieves robustness to abnormal tags but also dramatically improves the positioning precision. They provide detailed simulations to evaluate the proposed technique.

The next paper by S. Madhavi and T. H. Kim proposes a network model for the USN called a power control collision interference-free model and a distributed data aggregation scheduling protocol, which is adaptive to the rate and power. The authors test the protocol in a USN with different topologies and varying degrees. The results prove that the proposed power control collision interference-free model yields better results.

The paper by W. Z. Khan et al. surveys and classifies the existing detection schemes for nodes replication attack in wireless sensor networks. It comprehensively explores the various proposals in each category. The authors have taken a glance at some technical details and comparisons so as to demonstrate limitations of the existent detections as well as effective contributions.

The paper Ohood et al. performs a secure way based on biometric for WSN environment. The proposed protocol involves simple operations and light computations. The main advantage of the proposed protocol is using the user's iris to regenerate the user's key on-the-fly every time the user wants be authenticated which dramatically enhances security aspects in WSNs.

In the paper by S.-H. Hong et al. proposes a simple cluster-based data aggregation and routing algorithm (SCAR) that decreases the incurred overhead during the selection of cluster heads in wireless sensor networks. The performance results show that SCAR can prolong network lifetime via energy conservation and achieve energy balancing when nodes are fixed or have limited mobility.

In their paper, Yoo et al. propose a robust user authentication scheme, which fixes the security weaknesses of previous solutions and provides wider usability considering the use case when the sensor nodes cannot communicate with the gateway node. Once the solution is described, its security is ensured by formal proof and analysis against attacks. Additionally, performance and cost analysis are executed to determine its level of feasibility for real implementation.

The paper by X. Guo and Q. Luo presents a multisecret sharing scheme with application on multimedia sensor networks. The proposed scheme can reliably conceal private information. They perform security analysis and show that their proposed scheme provides secure, robust and trustworthy verification of dealer and participants over insecure wireless multimedia sensor networks.

In their paper, S. Ullah et al. present the key features of the IEEE 802.15.6 standard. The MAC, PHY, and security specifications of the standard are explained in detail. Different communication modes and access mechanisms are also presented. The narrowband (NB), ultra-wideband (UWB), and Human Body Communications (HBC) specifications are reviewed in terms of frame structure, modulation, and other important parameters. Finally, S. Ullah et al. highlight the security paradigm and services of the standard.

In their paper, G. Yang et al. propose a precision-enhanced and encryption-mixed privacy-preserving data aggregation (PEPDA). The objective is to reduce collision during data transmission and energy consumption and to compensate loss caused by the collision. Based on the Slice-Mix-AggRegaTe (SMART) scheme, it optimizes data slicing by using small data packet, node classifying, and positive and negative data slicing techniques. It also describes a randomized time slot and a data compensation algorithm. Theoretical analysis and simulation show that PEPDA demonstrates a good performance in terms of accuracy, complexity, and security.

In their paper, H. Wang and B. Yin perform image authentication in a wireless multimedia sensor network (WMSN). To verify the authenticity, creditability, and integrity of the received image in WMSNs, they propose a perceptual hashing-based robust image authentication scheme. The experimental results show that their scheme can provide compactness, visual fragility, perceptual robustness, and security.

In their paper, to remedy security flaws of Yeh et al.'s published protocol, W. Shi and P. Gong proposes a new authentication protocol for wireless sensor networks using elliptic curves cryptography. They perform security analysis and performance comparison and show that their improved scheme is more secure and efficient than the original one.

The paper by Ribeiro et al. proposes an address self-assignment protocol, which: uses negative acknowledgements and a flood control mechanism to minimize the energy spent; uses a technique named whispering to achieve robustness against malicious nodes; is able to detect dynamic network rejoint and dynamic node addition without exchanging specific messages; and handles both dynamic events without compromising routing tables.

The paper by W. Z. Khan et al. provides a mechanism for secure actor directed localization technique in wireless sensor and actor networks. It helps to locate a sensor node efficiently and effectively. Different security algorithms are tested to eliminate the possibility of attacks and registration of attacker nodes with other legitimate nodes in the network. The proposed technique prevents man-in-the-middle attacks and securely delivers data to the destination.

A paper of this special issue is contributed by Yu et al., in which they analyze security requirements on data collection and transmission from WSNs in the context of IoT and point out that it is imperative to resist node compromise attacks as well as DoS attacks without requiring the use of any per-shared keys. Therefore, they propose a data authentication and enroute filtering scheme, referred to as DAEF. They also analyze energy consumption to show the advantages of DAEF over some comparable schemes.

Muhammad Khurram Khan
Yang Xiang
Shi-Jinn Horng
Hsiao-Hwa Chen

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 126327, 10 pages
http://dx.doi.org/10.1155/2013/126327

Research Article

Secure Actor Directed Localization in Wireless Sensor and Actor Networks

Tamleek Ali,[1] Muazzam A. Khan,[2] Amir Hayat,[3] Masoom Alam,[3] and Muhammad Ali[1]

[1] *Institute of Management Sciences, Peshawar, Pakistan*
[2] *E&ME College, National University of Science and Technology, Islamabad, Pakistan*
[3] *COMSATS Institute of Information Technology, Islamabad, Pakistan*

Correspondence should be addressed to Masoom Alam; masoom.alam@gmail.com

Received 25 January 2013; Accepted 12 April 2013

Academic Editor: Muhammad Khurram Khan

Wireless sensor and actor networks are fully automated. Actor nodes are inducted to communicate with sensor nodes directly and reduce the communication delay caused by base station or sink nodes. Sometimes, the actor node is directly accessible without the involvement of any control room. The actor node is responsible for taking a prompt action against the reported event by a sensor node. For secure communication, it is essential that sensor and actor nodes be aware of their existing location and the data must be encrypted before transmission. Due to energy constraints, secure localization in wireless sensor networks is a hot issue. To date, the researchers have proposed many approaches for localization of sensor nodes in the network. In this paper, we provide new insights for secure actor directed localization technique in wireless sensor and actor networks. A secure connectivity based localization (CBL) approach for sensor and actor nodes localization is presented. The proposed approach helps to locate a sensor node efficiently and effectively. We have also decreased the possibility of attacks and the registration of attacker nodes with other legitimate nodes in the network. The proposed technique prevents man-in-the-middle attacks and securely delivers data to the destination.

1. Introduction

Wireless sensor network (WSN) is a collection of different sensor nodes which sense certain information from its surrounding and transfer it for further processing to one or more interested nodes called sinks. A number of WSN applications have emerged to observe an abnormal activity in the sensor's deployment area, such as military movement in a battlefield, sensing environmental changes, and health monitoring. In some applications, sending the state information only to a single node is not sufficient such as bomb blast and earth quake. Therefore, it is imperative to report to a group of nodes about the sensed events/data [1]. Due to resource limitations, sensor nodes usually adopt a multihop communication paradigm in which packets are passed to the destination through multiple intermediate nodes.

It is also sometimes necessary to respond to the sensed events/data by performing corresponding actions in that environment. For instance, in a fire handling system, the actors need to turn the water sprinklers on after receiving a report of fire. This leads to the emergence of wireless sensor actor networks (WSANs) [1, 2], which are a substantial extension of sensor networks, involving coexistence of sensors and actors in the same network. Thus, wireless sensor and actor networks (WSANs) are realized to enable the application to sense, interact, and change the physical world, for example, to monitor and manipulate the temperature and lighting in a specific area or the speed and direction of a mobile robot. It is envisioned that WSANs will be one of the most critical technologies for building the network infrastructure of future cyberphysical systems [3–5].

A WSAN is a networked system of geographically distributed sensor and actor nodes that are interconnected through wireless links. On receiving the required information, the actors make the decision about how to react to this information and perform corresponding actions to change the behavior of the physical environment. Sometimes there may be a base station which is responsible for monitoring and

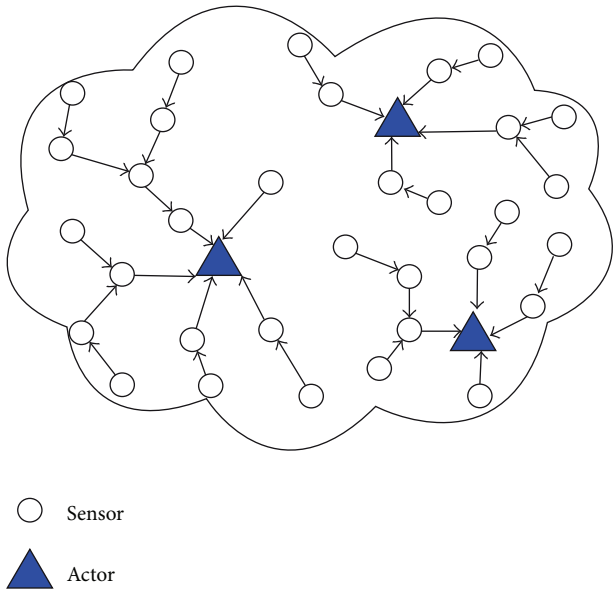


Figure 1: Wireless sensor and actor network.

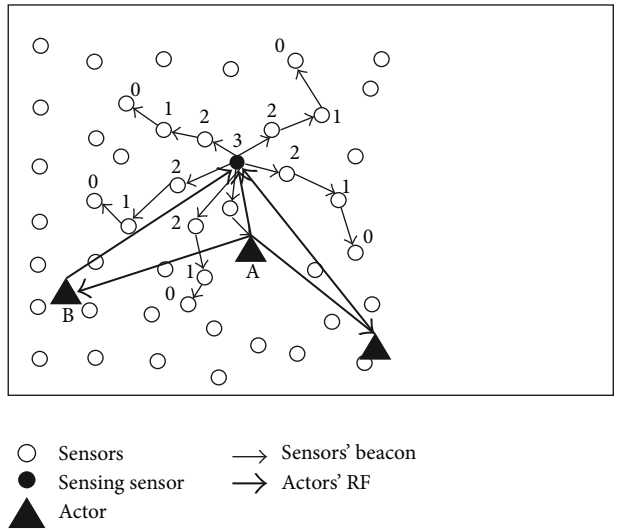


Figure 2: Actor sensor communication for localization.

managing the overall network through communications with sensors and actors. Figure 1 shows a view of wireless sensor actor network without any centralized control from a base station, but in most of the cases, there is a base station. For example, in case of fire anywhere in the network, the actor needs to turn on the water sprinkler to control the fire.

A WSAN can also monitor and manipulate the temperature and lighting in a smart office. In case of mobile robots, the actor may be able to change the velocity and direction of a mobile robot when there is any danger in a certain area [6]. Sensor nodes are generally stationary, whereas actors are mobile, for example, mobile robots and aerial vehicles. Figure 2 shows a typical scenario of actor sensor communication for localization where normal sensors are shown with their distances from sensing sensors. Sensors gather information about the state of the physical world and transmit the collected data to actors through single hop or multihop communication. On receiving the required information, actors make the decision about how to react to the event and trigger corresponding actions [7, 8].

Secure data delivery to actor nodes is very important for taking a prompt action at the right time and the right position. For data security, many encryption algorithms are proposed by researchers. These algorithms can be broadly divided into two types known as symmetric and asymmetric algorithms. In asymmetric cryptographic algorithms, two types of keys are used. Public key is used to encrypt data at the source node, while private key is used to decrypt data at the destination node. In symmetric cryptographic algorithms, there is a single key used for encryption and decryption. Both source and destination nodes should agree on that key in the start of communication. Therefore secure key distribution in symmetric algorithms is very important. Asymmetric algorithms are computationally very expensive as they need more energy and more computational power. Because these algorithms are based on high mathematical functions, they are not a suitable option for wireless sensor nodes which have lower computational power and energy. Keeping the keys secret and strong is very important in these algorithms; in case of using a weak key, the attacker can guess the used key after sniffing few data packets from the network. Once the used key is found, then the attacker can decrypt data, do any alteration, and also reencrypt it [7, 9].

Localization is a process to determine the exact locality of sensor nodes as well as actor nodes in the network. Location information of every node is very important for detection of an event to transfer any information and to record events. In a static network, it is easy to localize sensor nodes; however, in mobile networks as well as in large scale networks, manual localization is not possible. Similarly, to provide the equipment of GPS on each sensor node is quite expensive in terms of cost and energy consumption [10, 11].

The localization algorithms are divided into two categories:

(a) range based,

(b) range free.

In range based algorithms, specialized hardware is used to estimate their distance from seeds. However, range free algorithms do not use any specialized hardware, radio signal strength, angle of signal arrival, or distance measurement. Range free algorithms require that each node knows the nodes inside its radio range, estimated locations, and the ideal range of each sensor node. However, due to insecure communication between sensor and actor nodes, any attacker node can take part in this communication and later on become the legitimate member of the network. To avoid such types of attacks in Khan et al.'s [10] work, we proposed a secure localization technique where sensor and actor nodes will first encrypt their data with an efficient predecided algorithm. This technique will prevent the attacker nodes registration as well as intervention in communication among sensor and actor nodes.

In this paper, we proposed a secure connectivity based approach to localize sensor and actor nodes in the network as well as to prevent the attacker nodes from joining the group and becoming a legitimate node of the network. The proposed localization approach works efficiently in both random and precise deployed sensor nodes. The rest of the paper is organized as follows. Section 2 presents related work with different localization approaches proposed to date in the research community. Section 3 explains our proposed approach in different scenarios (an emergency scenario with normal communication and attacker scenario where an attacker node tried to get registered). Section 4 presents our simulation results and discussion. Section 5 which is the last section of this paper has the conclusion and future work.

2. Related Work

In this section, we discuss the literature survey about localization. We explain different localization protocols proposed by researchers for localization in wireless sensor networks to date. There are two main categories of these protocols: range based, which works in a limited communication range, and the range free approach, which has no such restrictions. These categories are further discussed here.

2.1. Range Free Localization. Range free localization schemes localize the sensor nodes in the network with the help of radio connection information shared with their neighbor nodes or with their ability to sense the environment which each sensor node can do easily. There are many range free approaches for localization; we can divide them into two subcategories on the basis of their characteristics:

(i) anchor based schemes, where some special nodes called anchor nodes are required in the network and have the ability to know their location

(ii) anchor free schemes, where no special anchor nodes are required for localization.

The most important aspect of range free localization is its implementation on simple sensor node instead of high-cost sensor nodes equipped with specialized hardware. Similarly, it has the same drawbacks of range based localization like radio propagation that is area and time dependent, attenuation, interference and multipath propagation causes anchor based localization schemes more costly and expensive.

He et al. [12] in 2009 proposed a new approach to localize a lost node or newly registered node in the network, and it was later improved by Wang and Jin [13]. In this scheme, a new node that is entered or may not be in range initially receives information from its neighbor nodes and then each node is connected with another node to make a triangle; every node checks either it is in a triangle or not to calculate the center of gravity (COG) of the triangle, through which a node is able to estimate its position. APIT also performs better when node placement is random as well as with irregular radio patterns. Teng et al. [14] in 2009 developed mobile beacon assisted localization (MBL) for wireless sensor networks. In this approach, all nodes are unknown and they have no information about their surrounding nodes and their own locations. Initially, for each unknown node, a set of samples is chosen randomly from the whole deployment area. A set of uniformly distrusted samples are used to represent those sets having equal weights. When the weight of a set is equal to 1, it actually represents the importance of that corresponding sample, which estimates the location of the unknown node. The current position of a beacon which may be the initial position is also chosen randomly from the whole deployment area.

Rudafshani and Datta [15] proposed new algorithms for mobile as well as for static wireless sensor networks called MSL and MSL*. Both MSL and MSL* algorithms are able to handle heterogeneous networks in a radio transmission range. MSL* maintains a list of probable locations (samples) as the location information of each node in the network, and then weight is assigned to each sample that estimates its gravity. The weight of a sample is an approximation that represents the true location of a node. In MSL*, communication is the transfer of samples between nodes, and so each node uses the sample of its neighbors for the calculation of its own samples. While in MSL they assign a weight to each node for computation of its own samples, each node uses the weight of nodes in its surrounding. After the computation of weights, MSL is responsible for estimating a single location and a closeness value. Then each node broadcasts its estimated value and closeness to its neighbor nodes.

Ma et al. [16] for the first time proposed a secure localization technique for wireless sensor actor networks (WSANs). These networks are different from simple wireless networks due to nodes heterogeneity. This approach is based on DV-Hop (the most basic scheme which employs a classical distance exchange so that all nodes in the network get distance) and hidden actors, where actor nodes are responsible for locating a sensor node in the networks. The actor node continuously receives authentication messages and minimum hop numbers from sensor nodes, and then the nearest actors collectively compute the location of sensors through actor-actor communication and maximum likelihood estimators MLE (the parameters that maximize the probability (likelihood) of the sample data). Xiao et al. [17] proposed a novel scheme for localization in irregular areas of wireless sensor networks called reliable anchor based localization (RAL). There are three important characteristics of this approach: (i) it uses an average hop length estimation algorithm, which provides tolerance against irregular radio propagation and distortion effects due to many obstacles in an irregular area, (ii) achieves high accuracy due to information received from reliable anchors, and (iii) it introduces virtual anchors. All the anchors are equipped with GPS and have the ability to know their positions. There are four basic steps in RAL: (i) propagation of anchor information in the network, (ii) calculation of average hop length and reliable anchors, (iii) determination of the area of sensors, and (iv) determination of an anchor for sensor.

DRFL uses a cluster based approach for localization of nodes in a wireless sensor network; it was proposed in 2007 by Qiu and Xu [18]. The clustering is a data mining technique which actually separates the studied objects on the basis of

distance from each other into different groups. The clustering technique also helps in security by excluding the bad possible positions of nodes, which improves the localization accuracy. A new distributed localization scheme for wireless sensor networks was proposed by Kuang et al. [19] in 2008 known as VB-ERL. Using this scheme, all nodes in the network are static except for a few nodes which can move from one location to another location. These mobile nodes use virtual beacons to broadcast their location information in the network. Each sensor node receives that beacon and estimates its own location on the basis of received information using the proposed algorithm. Mobile nodes move in the network through Guass Markov mobility model and broadcast their location information in the network.

Shah and Khan [20] proposed a time based localization approach in wireless sensor actor networks. They introduced a new algorithm called timing based mobile sensor localization (TMSL) algorithm, where actor nodes are used for localization of sensors in the network which broadcast beacons after specified intervals, and after receiving that beacon, each sensor node is responsible for using its propagation time and speed of RF signal for the calculation of its distance from actor node. These beacons have certain information like propagation time and actor identity.

2.2. Range Based Localization. Range based localization is hardware dependent. In range based approaches, each node has to estimate its point-to-point distance and angle from its neighbors. In such type of localization, the researchers mostly assume that, from signal strength, the distance between sender and receiver can be estimated or the time of flight of the data from sender to receiver can help in this regard. In contrast to range free approaches for localization which are independent of all these restrictions, here, the sensor nodes are required to be equipped with extra hardware and coordination. Energy consumption is also higher and thus overall cost of the network having a range based localization technique is higher as compared to a range free technique [21]. Han et al. [22] in 2005 proposed an efficient cooperative localization scheme for wireless sensor actor networks. The ECLS scheme is totally dependent on an event in the network because it is an event driven scheme.

The ECLS scheme works on actors' cooperation. In this scheme, the actor nodes use GPS or some other mechanisms to know their real-time location in the network, which is possible, as we have discussed in the introductory part of this paper, because these actor nodes are more powerful and have no energy constraints. After getting its position, the actor node floats its position in the network as a beacon; the mobile actor nodes actually serve as replacements for anchor nodes in the wireless sensor network. Here the lifetime of beacons is defined and every node which receives the beacon reduces it by one and forwards it to its neighbors. When its lifetime gets to 0, any node which receives this beacon will discard it. The nearest actor will receive the beacon and that will directly communicate with other actors to take part in the localization process.

Wang et al. [23] proposed a direction based localization scheme (DLS) for sensor networks. DLS uses multiple beacons to determine the direction of a sensor node. In this scheme, some anchor nodes are also deployed which work as reference nodes. For efficient and precise position measurement, DLS uses a virtual dual direction coordinate (VDDC) system. DLS estimates the direction of each sensor node on the basis of data dissemination and direction of its neighbor nodes in the network. The sink node that lies normally at the center of the network and works as a control center is responsible for originating the localization requests in the network. Therefore, DLS easily and correctly identifies the direction of all sensor nodes that are nearest to the sink node. The sink node initiates the DLS by sending a locating request (LREQ) to all neighbor nodes, and only anchors have the responsibility to send LREQ packets with their directions. However, no sensor node has to do this. When a sensor node receives an LREQ packet from an anchor in the surrounding area, it estimates its direction according to the direction information in the LREQ packet and propagates its direction information to the surrounding sensors.

Li et al. [24] in 2008 worked on localization of sensitive service discovery in wireless sensor networks. In iMesh, an information mesh is constructed by a service provider that is actually used as a localized planner structure. For this purpose, a blocking rule is used that is further enhanced with a newly proposed expansion rule. iMesh-A is the basic version of iMesh which only uses the information blocking rule, whereas the complete protocol consisting of the blocking rule and the expansion rule is called iMesh-B. In iMesh-B, the localization information is published by the service provider about all directions in the mesh like east, west, north, and south. The node may receive information from multiple nodes, but it only forwards the information from the closest nodes.

In 2008, Won and Song [25] proposed a new technique for localization, that is, anchor free localization. For this purpose, the authors used a map to map stitching approach. In this approach, the whole network is divided into small subregions that overlap each other and each creates a local map that is further refined through an optional refinement phase which improves the quality of these maps. These local maps are stitched together to form a global map; however, the map stitching order strongly influences the localization performance. Shang et al. [26] used connectivity information for localization of sensor nodes in 2004. Every sensor node is able to know its neighbor nodes which are in its communication range. This approach is based on multidimensional scaling (MDS). MDS is a data analysis approach that transforms proximity information into geometric embedding and takes advantage of distance information between nodes that have yet to be localized. MDS is very suitable for communication networks where localization of nodes is done through distance information among different nodes using their coordinates in two-dimensional and three-dimensional areas. There are two major parts of this approach: (i) MDS-MAP(C) that constructs a global map for localization in the network and (ii) MDS-MAP(P) that constructs many small maps throughout the network and then combines them

together into a global map. There are three basic steps which started with local distance measurement: (i) measure the shortest path distance of each node with other nodes, (ii) derive nodes' coordinates using MDS, and (iii) normalize the resulting coordinates to take into account any nodes whose positions are known [27, 28].

Lazos et al. [29] address the problem of enabling sensors of WSN to determine their location in an untrusted environment. As localization schemes based on distance estimation are expensive for the resource constrained sensors, they proposed a range independent localization algorithm known as SeRLoc. It was a distributed algorithm and did not require any communication among sensors. Lazos and Poovendran [30] proposed secure and robust localization. They identified the problem of verifying the location claim of a node, known as location verification, in wireless sensor networks (WSNs). Their robust positioning system, called ROPE, enables sensors to determine their location without any centralized computation. It also provides a location verification mechanism that verifies the location claims of the sensors before data collection. Li et al. [31] identified some unconventional security threats, which rather adversely affect the ability of localization schemes to provide trustworthy location information. They identified a list of attacks that were unique to localization algorithms. Since these attacks were diverse in nature and there were some unforeseen attacks, therefore, they could bypass traditional security countermeasures. They developed robust statistical methods to make localization attack tolerant. Their solution worked on two broad classes of localization, that is, triangulation and RF based fingerprinting methods. Liu et al. [32] proposed two methods to tolerate malicious attacks against beacon based location discovery in sensor networks. One method filters out malicious beacon signals on the basis of the "consistency" among multiple beacon signals, while the second method supposedly tolerates malicious beacon signals by adopting an iteratively refined voting scheme. Both methods were authentication independent and provided that the benign beacon signals constitute the majority of the "consistent" beacon signals. Jadliwala et al. [33] proposed secure distance based localization in the presence of cheating beacon nodes in mobile wireless ad hoc and sensor networks. They identified the problem of wrong location estimation and proposed a threshold value above which error free estimation would not be possible. On the bases of these values, they proposed secure localization algorithms. Attacks on the localized mobility are identified in RFC4832 [34]. It identified the impersonation and man-in-the-middle attacks. Wang et al. [35] developed a new landmark selection algorithm with incremental Delaunay refinement method that does not assume any knowledge of the network boundary and runs in a distributed manner to select landmarks incrementally until the global rigidity property is met [36, 37].

3. Emergency Scenarios

In this section, we discuss our proposed scheme for localization in rescue operations after huge destruction where we deploy sensor network. The sensor nodes are uniformly deployed; we assume medium nodes density in the network, where actor nodes play the role of anchors. The proposed scheme works on coordination among actors in the network. We assume two actors (anchors) in the network having greater signal strength, power, and energy as compared to sensor nodes. There are two anchor nodes in the network used for localization of sensor nodes.

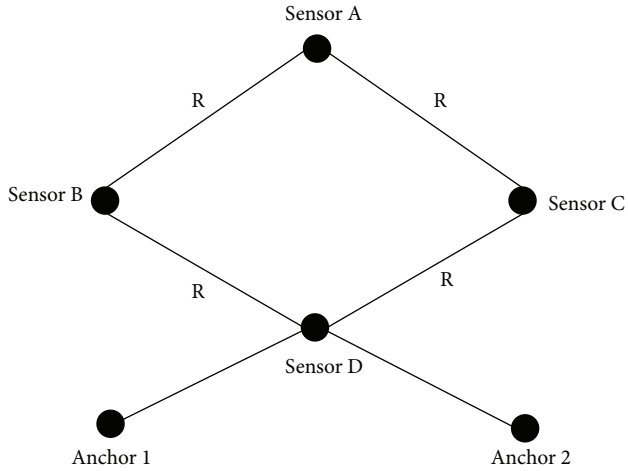


Figure 3: Simulation scenario.

3.1. Example Scenario. Figures 5 and 6 shows the localization scenarios of two and three nodes, respectively. In the first case, the two anchor nodes will flood their location information in the network. The sensor nodes closest to the anchor nodes will receive information from both anchors. The sensor node D will calculate its position on the basis of received information. Sensor nodes at far position will receive that packet through sensor node D, which will increment the packet hop count by 1. Any sensor that will receive the anchor information packet will increment the hop count and every node is able to know its position, neighbor nodes, and its distance from anchor node. If a sensor node receives an information packet from two different anchors it will join the closest anchor node. If the distance for both anchors is the same and no alternative route is available without that sensor node to access other nodes of the network as in Figure 3, then the sensor node will work as a router and forward the node for both anchors.

Figure 3 explains our simulation scenario where many nodes are deployed and they donot know their location. Therefore the initiative step will be taken by the two anchor nodes to broadcast their information and provide a starting point for sensor nodes to compute their position with respect to the anchor [9, 10].

Figure 4 shows the process when node D received information packet from both anchors 1 and 2; as discussed earlier, there is no option without node D to access sensor nodes A, B, and C. Therefore sensor node D will be responsible for forwarding anchor information to other nodes. The process will continue till all of the sensor nodes join the network and know their position as well as the position of anchors.

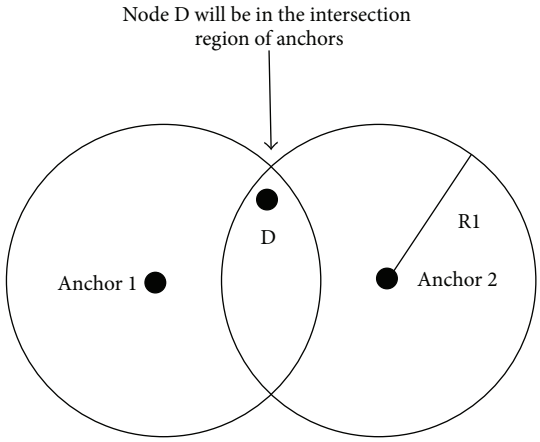


Figure 4: Connectivity of anchors with node D.

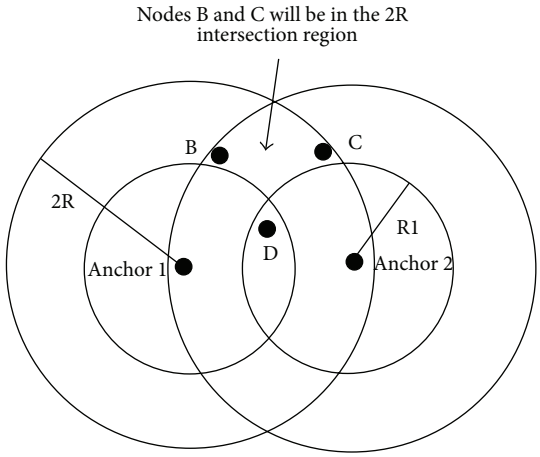


Figure 5: Localization of nodes B and C using node D.

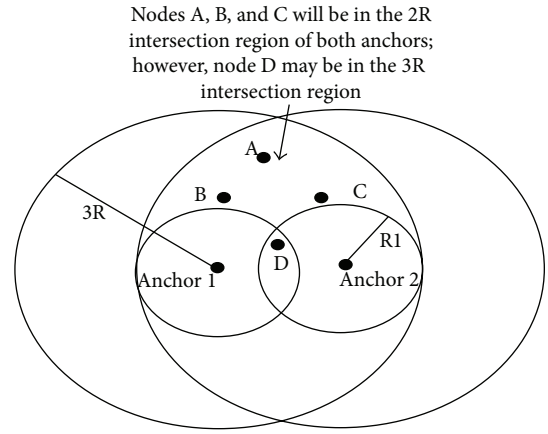


Figure 6: Localization of node A with nodes B and C.

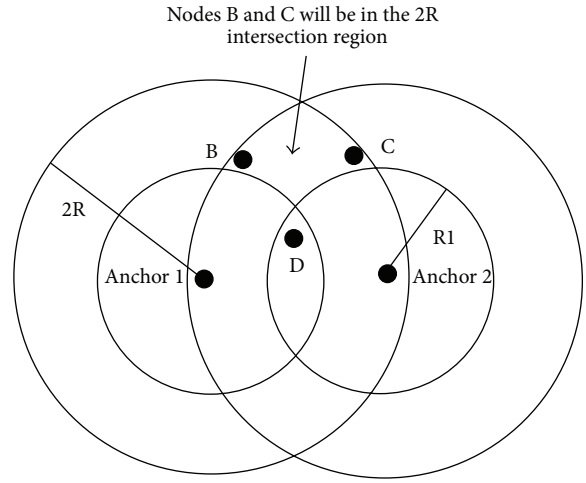


Figure 7: Attacker scenario (node D is the attacker node).

However the drawback of this approach was that many unwanted and attacker nodes could be able to join the network.

3.2. Attacker Scenario. Now we discuss an attacker scenario where the attacker node will send a join request to an anchor node. The anchor node will consider it as a legitimate node and will allow him it communicate and use the network resources because there is no mechanism to authenticate a node or to check the authorization whether the node is trustworthy to join or not. Figure 7 explains a scenario where node D is an attacker node. Due to the absence of any security mechanism in the network, node D gets registered with the anchor nodes and the other nodes in the network also consider node D as a legitimate node and join it as a forwarding/routing node to access anchor nodes.

To prevent the registration of these attackers, we use a symmetric key approach for authorization and authentication of our trusted sensor nodes. Before deployment of anchor and sensor nodes, we should provide them with a predefined key for authentic communication as well as to maintain data integrity and confidentiality. It provides end to end communication security among sensors and anchor nodes. Although this technique is in unable to handle man-in-the-middle attack, therefore, we still need a mechanism for prevention of man-in-the-middle attack. Such type of attacks can misguide our network system; the attacker can inject faulty routes in the routing table and replace the data packets and so forth.

Therefore this technique is not suitable for sensitive regions like atomic reactors and battlefields where man-in-the-middle attack can cause a big destruction. To achieve high level of security, we introduce traditional security mechanism, in which RC-6 algorithm is used for data encryption. As it consumes high energy as compared to symmetric key, however, it also provides high level security. There are many scenarios where we could not compromise security as discussed earlier.

4. Results and Discussion

In this section we discuss probabilities of the location of different nodes using our simulation scenario from Figure 3.

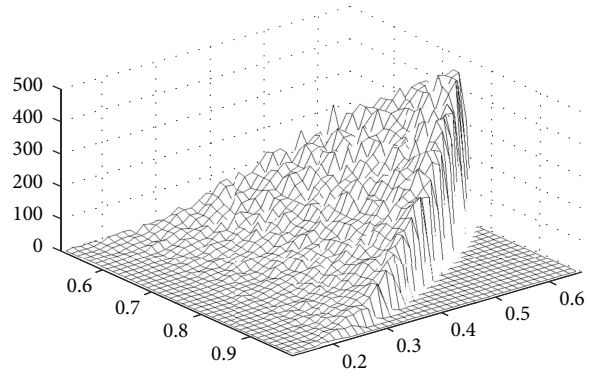


Figure 8: Probabilities for node D location.

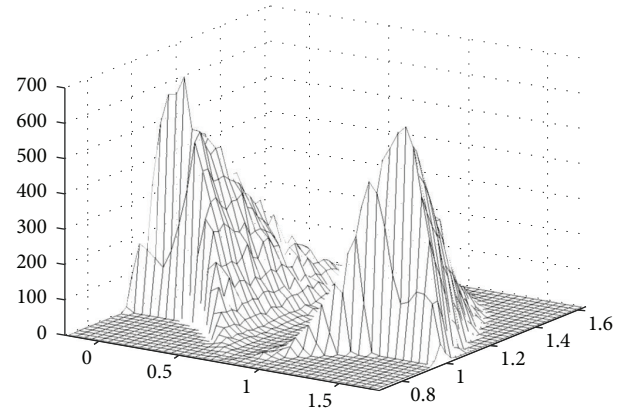


Figure 9: Probabilities for nodes B and C location.

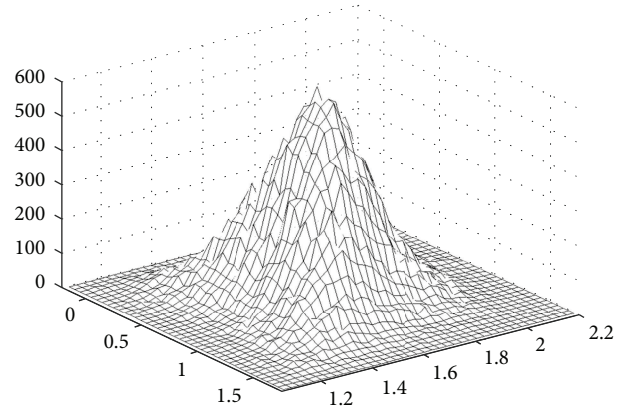


Figure 10: Probabilities for node A location.

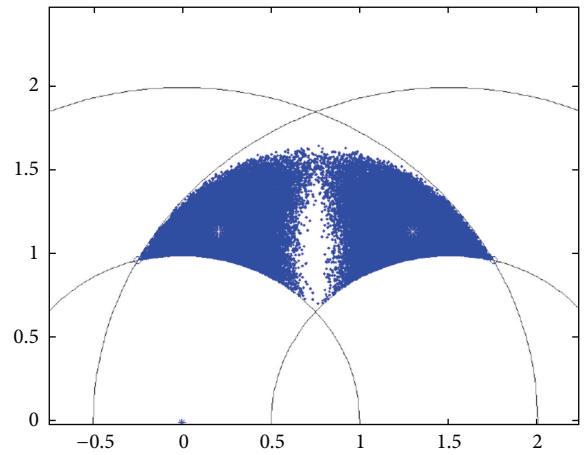


Figure 11: White dots in the middle of the blue area represent the location of node B and node C.

These are found by generating random node locations and finding the frequency with which nodes are located in certain positions when they satisfy the topology constraints.

Figure 8 shows the number of times node D was located in certain positions. If normalized, this could also be viewed as a probability distribution. The probability of node D location increases in an upward direction, depending on the other nodes position as well as anchor nodes. Node D is directly connected with both anchor nodes as well as with nodes B and C.

The probabilities of nodes B and C are shown in Figure 9. We assume that nodes B and C are far away from each other that they are not directly connected. Therefore it is also clear from their probabilities diagram that there is some space between these nodes.

Figure 10 shows the probabilities of node A location in our particular scenario. Node A is directly connected with both nodes B and C; therefore, the graph is denser in the middle as compared to its edges. Figure 10 shows the position of node A that lies in the middle of both anchors 1 and 2. After localization of node D, it helps to localize nodes B and C.

Figure 11 gives another way to view the position of node B on the left side of the diagram and node C on the right side. There is a certain gap between B and C which clarifies that these nodes cannot be so close to each other.

The diagram in Figure 12 explains the positions of both nodes D and A in their particular locations.

The mean and standard deviation (a measure of the dispersion of a set of data from its mean) values for the locations of the nodes are as follows:

Node A: (0.7493, 1.6138) σ = (0.2942, 0.1610),

Node B: (0.2045, 1.1404) σ = (0.1743, 0.1292),

Node C: (1.2949, 1.1404) σ = (0.1743, 0.1292),

Node D: (0.7499, 0.4684) σ = (0.0831, 0.0966).

The results shown above make it clear that actor nodes are very important and the localization procedure starts from actor and then onward responsibilities are shared with other nodes. From our simulation results, it is clear that beacon broadcasting by actors provides a base to sensor nodes for the computation of their position. This approach shifted the computational overhead on the actor node having high resources, and therefore this approach reduces energy consumption as well as increases networks lifetime.

The graph in Figure 13 explains memory requirements for secure approach and nonsecure approach. From the graph it is clear that as we increase the level of security, it also increases

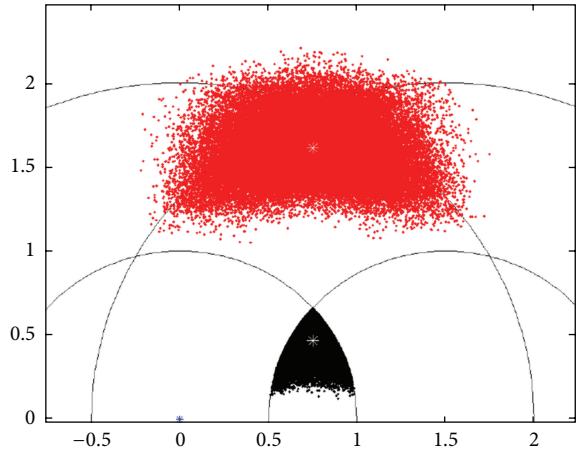


Figure 12: White dots in the middle of the red and black areas represent the location of node A and node D.

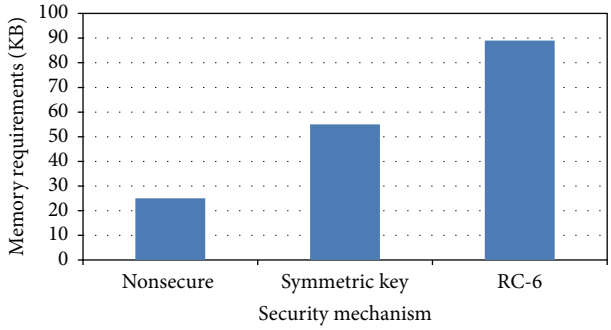


Figure 13: Memory requirements (KB) of different mechanisms.

the memory requirements at sensor nodes. RC-6 provides a high level of security as compared to the other two approaches and that is why it has high memory requirements whereas the other two approaches need lower memory than RC-6.

Figure 14 presents a relationship between compromised sensor nodes and network node density. As long as node density in the network increases, number of compromised sensor nodes also increases. However we can control the number of compromised sensor nodes with the help of a strong security mechanism. RC-6 based security mechanism minimizes the number of compromised sensor nodes in the network, while the other two mechanisms are unable to control compromised sensor nodes and gradually increase with the increase in node density.

The relationship between number of successful attacks and time is shown in Figure 15. As time passes, the attackers are successful in intruding into the network and getting registered with anchor and sensor nodes. However using a good security mechanism can reduce the number of attacks. In Figure 15 the number of successful attacks is very high in nonsecure approach, while induction of security mechanism reduces these attacks. In case of symmetric key, number of attacks is lower and minimized much more using RC-6 based security.

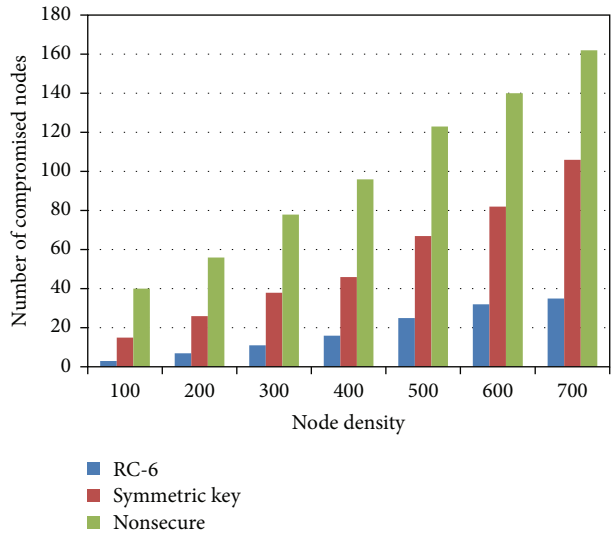


Figure 14: Node density versus number of compromised sensor nodes.

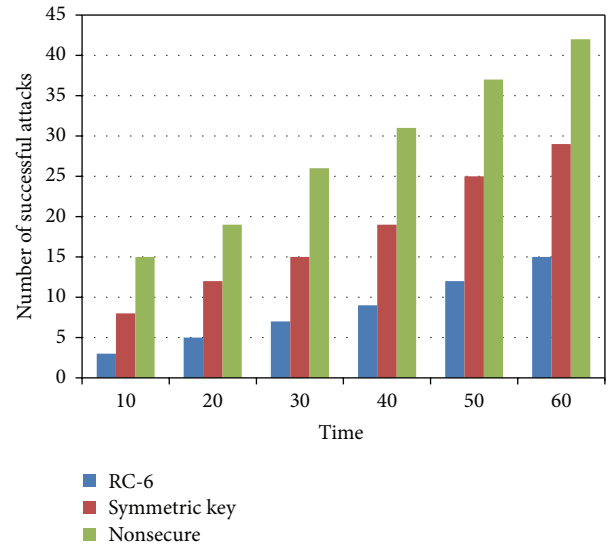


Figure 15: Number of Successful Attacks versus Time.

Figure 16 presents the impact of security mechanism on network lifetime. It is obviously clear that security algorithms increase computational time of sensor nodes and utilize more energy than a normal procedure. Therefore security strength is directly proportional to the network lifetime. High level of security will decrease the network life time because it involves more energy utilization. The graph given here also explains the same relationship; RC-6 provides high level security and it reflects back on the network life. Whereas symmetric and other mechanisms have lower energy consumption, the network life is greater than in RC-6 based networks.

Figure 17 shows a relationship between number of unknown nodes and percentage error. In case of older approaches, ADO and MSL, the percentage error rate is higher than in our proposed approach (CBL). The main

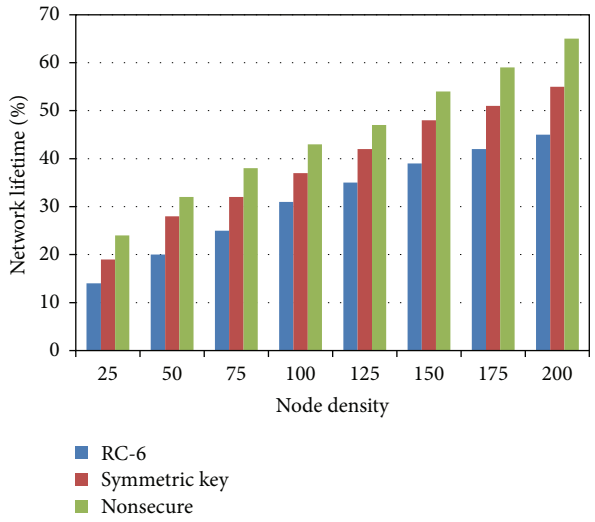


Figure 16: Network lifetime versus node density.

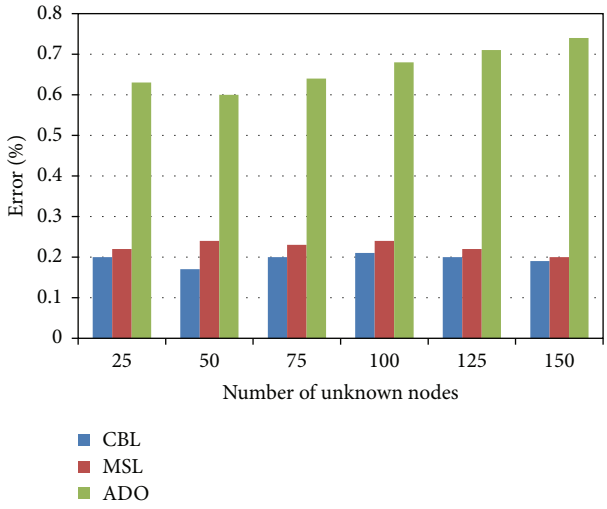


Figure 17: Error due to unknown nodes.

reason is that anchor node is able to track a sensor node due to its high transmission and receiving power than the other two approaches. An anchor node can handle the movement of nodes as well as any hurdle in the transmission.

5. Conclusion and Future Work

In this paper we proposed a secure mechanism for localization of sensor nodes in wireless sensor networks. Using an encryption algorithm for secure data delivery and registration of sensors with anchor node, we effectively minimize and block the external attacks. After simulation results, we conclude that efficient localization in sensor networks can be greatly enhanced by the understanding of both connectivity of sensor nodes and to which nodes they are not connected. The mechanism shows a particular area in which a node can be localized, and we can easily find it there. Once the anchor node locates its own position, the sensor nodes are able to localize each other. This approach is initiated by the anchor node having higher resources than sensor node; therefore, it will reduce energy consumption as well as increase networks lifetime. However the future work is to stop the internal attacks and reduce the number of compromised sensor nodes in the network.

References

[1] F. Xia, W. Zhao, Y. Sun, and Y.-C. Tian, "Fuzzy logic control based QoS management in wireless sensor/actuator networks," *Sensors*, vol. 7, no. 12, pp. 3179–3191, 2007.

[2] F. Xia, Y.-C. Tian, Y. Li, and Y. Sun, "Wireless sensor/actuator network design for mobile control applications," *Sensors*, vol. 7, no. 10, pp. 2157–2173, 2007.

[3] A. Rezgui and M. Eltoweissy, "Service-oriented sensor-actuator networks," *IEEE Communications Magazine*, vol. 45, no. 12, pp. 92–100, 2007.

[4] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: Research challenges," *Ad Hoc Networks*, vol. 2, no. 4, pp. 351–367, 2004.

[5] "NSF Workshop on Cyber-Physical Systems, Research Motivation techniques and roadmap," October 2006, http://varma.ece.cmu.edu/cps.

[6] M. A. Khan, G. A. Shah, and M. Sher, "A survey of multicast routing in wireless sensor networks," in *Proceedings of International Conference of World Academy of Science, Engineering and Technology Conference*, Amsterdam, The Netherlands, September 2009.

[7] M. A. Khan and A. Salam, "Evaluation of different cryptographic algorithms for wireless sensor and actor networks (WSANs)," *Wulfenia Journal*, vol. 19, no. 10, 2012.

[8] M. A. Khan, A. Rehman, M. Zakaya, and G. A. Shah, "Challenges for security in wireless sensor networks (WSNs)," *Journal of Computing*, vol. 4, no. 9, 2012.

[9] M. A. Khan, C. Beard, and M. Sher, "A comparison of different localization approaches in wireless sensor networks," in *Proceedings of International Conference on Computer Communications and Networks (CCN '10)*, Orlando, Fla, USA, July 2010.

[10] M. A. Khan, M. Sher, and C. Beard, "Connectivity based localization approach for wireless sensor actor networks," in *Proceedings of International Conference on Electrical, Computer, Electronics & Communication Engineering*, pp. 21–24, Penang, Malaysia, February 2011.

[11] M. A. Khan, G. A. Shah, M. Ahsan, and M. Sher, "An efficient and reliable clustering algorithm for wireless sensor actor networks (WSANs)," in *Proceedings of the 53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS '10)*, pp. 332–338, August 2010.

[12] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range free localization schemes for large scale sensor networks," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 81–95, San Diego, Calif, USA, September 2003.

[13] J. Wang and H. Jin, "Improvement on APIT localization algorithms for wireless sensor networks," in *Proceedings of International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '09)*, pp. 719–723, April 2009.

[14] G. Teng, K. Zheng, and W. Dong, "Adapting mobile Beacon-assisted localization in wireless sensor networks," *Sensors*, vol. 9, no. 4, pp. 2760–2779, 2009.

[15] M. Rudafshani and S. Datta, "Localization in wireless sensor networks," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 51–60, Cambridge, Mass, USA, April 2007.

[16] J. Ma, S. Zhang, Y. Zhong, and X. Tong, "SeLoc: secure localization for wireless sensor and actor network," in *Proceedings of International Conference on Mobile Ad Hoc and Sensor Sysetems (MASS '06)*, pp. 864–869, October 2006.

[17] B. Xaio, L. Chen, Q. Xaio, and M. Li, "Reliable anchor based sensor localization in irregular areas," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 60–72, 2009.

[18] M. Qiu and H.-M. Xu, "A distributed range-free localization algorithm based on clustering for wireless sensor networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '07)*, pp. 2633–2636, September 2007.

[19] X.-H. Kuang, H.-H. Shao, and R. Feng, "New distributed localization scheme for wireless sensor networks," *Acta Automatica Sinica*, vol. 34, no. 3, pp. 344–348, 2008.

[20] G. A. Shah and O. B. Khan, "Timing based mobile sensor localization in wireless sensor actor networks," in *Mobile Networks and Applications*, pp. 664–679, Springer, Amsterdam, The Netherlands, 2010.

[21] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom '01)*, pp. 166–179, July 2001.

[22] P. Han, C. Gao, M. Yang, D. Mao, and B. Yu, "ECLS: an efficient cooperative localization Scheme for wireless sensor and actor networks," in *Proceedings of the 5th International Conference on Computer and Information Technology (CIT '05)*, pp. 396–400, September 2005.

[23] S.-S. Wang, K.-P. Shih, and C.-Y. Chang, "Distributed direction-based localization in wireless sensor networks," *Computer Communications*, vol. 30, no. 6, pp. 1424–1439, 2007.

[24] X. Li, N. Santoro, and I. Stojmenvoic, "Localized distance-sensitive service discovery in wireless sensor networks," in *Proceedings of the 1st International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing (FOWANC '08)*, Hong Kong, May 2008.

[25] O.-H. Won and H.-J. Song, "Localization through map stitching in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 1, pp. 93–105, 2008.

[26] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 961–974, 2004.

[27] B. Xiao, H. K. Chen, and S. G. Zhou, "A walking beacon-assisted localization in wieless sensor networks," in *Proceedings of IEEE International Conference on Communications (ICC '07)*, pp. 3070–3075, Glasgow, Scotland, June 2007.

[28] M. Rudafshani and S. Datta, "Localization in wireless sensor networks," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 51–60, Cambridge, Mass, USA, April 2007.

[29] L. Lazos, R. Poovendran, and S. Čapkun, "ROPE: robust position estimation in wireless sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 324–331, April 2005.

[30] L. Lazos and R. Poovendran, "SeRLoc: secure range-independent localization for wireless sensor networks," in *Proceedings of the ACM Workshop on Wireless Security (WiSe '04)*, pp. 21–30, October 2004.

[31] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 91–98, April 2005.

[32] D. Liu, P. Ning, and W. K. Du, "Attack-resistant location estimation in sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 99–106, April 2005.

[33] M. Jadliwala, S. Zhong, S. Upadhyaya, and C. Qiao, "Secure distance-based localization in the presence of cheating Beacon nodes," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 810–823, 2010.

[34] C. Vogt and J. Kempf, "Security threats to network-based localized mobility management (NETLMM)RFC-4832," Network Working Group, Category: Informational, The IETF Trust, April 2007.

[35] Y. Wang, S. Lederer, and J. Gao, "Connectivity-based sensor network localization with incremental delaunay refinement method," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 2401–2409, April 2009.

[36] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'," *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.

[37] M. K. Khan, "Fingerprint biometric-based self-authentication and deniable authentication schemes for the electronic world," *IETE Technical Review*, vol. 26, no. 3, pp. 191–195, 2009.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 693519, 16 pages
http://dx.doi.org/10.1155/2013/693519

Research Article

A Framework for Robust Address Assignment in WSNs Whispering to Avoid Intruders

Carlos Ribeiro, Ivo Anastácio, André Costa, and Marcia Baptista

INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, R. Alves Redol 9, 1000 Lisboa, Portugal

Correspondence should be addressed to Carlos Ribeiro; carlos.ribeiro@ist.utl.pt

Received 22 October 2012; Revised 2 August 2013; Accepted 4 August 2013

Academic Editor: Yang Xiang

Wireless sensor networks (WSNs) are becoming bigger, and with this growth comes the need for new automatic mechanisms for initializations done by hand. One of those mechanisms is the assignment of addresses to nodes. Several solutions were already proposed for mobile ad hoc networks but they either (i) do not scale well for WSN; (ii) have no energy constraints; (iii) have no security considerations; (iv) or have no mechanisms to handle fusion of network partitions. We proposed an address self-assignment protocol which uses negative acknowledgements and an improved version of a flood control mechanism to minimize the energy spent; uses a technique named *whispering* to achieve robustness against malicious nodes; is able to detect dynamic network rejoint and dynamic node addition without exchanging specific messages; and handles both dynamic events without compromising routing tables.

1. Introduction

Wireless sensor networks (WSNs) have been arousing the interest of both researchers and the general community. WSNs are networks composed of small and cheap devices with sensing abilities which are able to communicate with each other through radio signals. The combination of sensing and radio communication abilities makes these networks ideal to build distributed sensing networks where each node collaborates by sensing one or more phenomena in its neighborhood and relaying it to a central node.

In order to be cheap and last for long periods without management, sensor nodes have several challenging constraints, from which the most important one is energy. Thus, every algorithm and protocol designed for sensor networks should always be energy conservative.

Given that sensor networks should be deployed on every kind of environment, including very hostile environments, security should be a major concern. Usually, achieving security implies some energy loss. However, this loss should be kept to a minimum when there is no threat to defend against.

1.1. The Naming Problem. One of the problems of sensor networks is the naming. Given that a sensor network could be comprised of a large amount of nodes, the unique addressing of each node may be a problem. Currently, nodes are initialized by hand with a unique number when the code is uploaded to the sensor node. In the initial versions of wireless sensors' operating systems, every sensor had to be programmed individually through physical contact using a special programming device. In those days, initialization was not a big issue because it could be easily done with the programming. However, currently, wireless sensors are being programmed using their wireless network [1, 2], which makes naming much more difficult.

Given that sensor programming is currently done by wireless radio and that wireless radio communications require addressing each individual sensor, the naming cannot be piggybacked on sensor programming as it used to be.

The 6LowPan [3] initiative solves this problem by assigning an IPv6 address to each node, which ensures its uniqueness because it results from the combination of the personal area network (PAN) address, where the node is, and the 64 bit unique manufacturer address of the node's link layer [4]. Since these data are known by each node, the assignment can be done without contacting the neighborhood, thus achieving a zero configuration solution. However, 6LowPan assumes the existence of a link layer address and depends on its

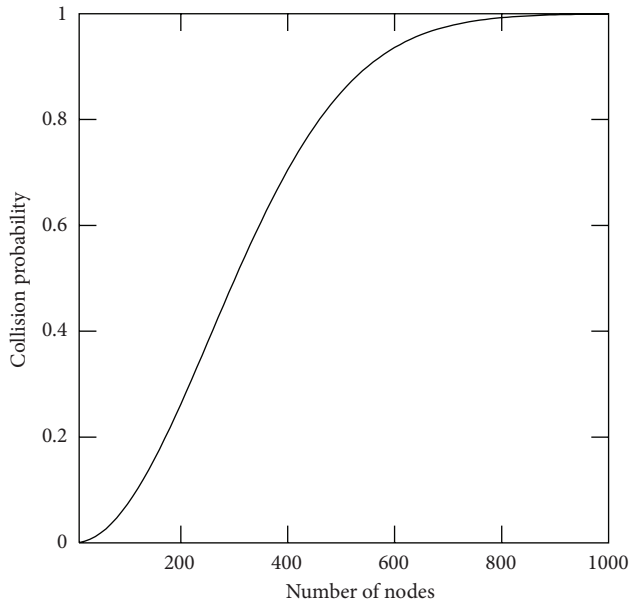


Figure 1: Collision probability with the number of nodes deployed.

existence. 6LowPan is specially tuned to be used with the 802.4.15 link layer which supports two types of address: a 64 bit address and a 16 bit address. The 64 bit address is a global unique manufacturer address, and the 16 bit address is a dynamic address unique only within each PAN. The 16 bit address is used whenever possible because 64 bit addresses make the communication headers too big for small devices. In TinyOS, the usual payload length is 29 bytes, and the maximum packet size in 802.15.4 radio is 128 bytes [5]; thus, it is very inefficient to use 16 bytes just for the link layer addressing purposes (8 bytes for the receiver and 8 for the sender).

The 802.15.4 [6] 64 bit unique address is manufacturer specified; thus, it is always available, but the 16 bit address must be derived by a special protocol that ensures its uniqueness within the PAN. Notice that it is not possible to use the random assignment solution, as in IPv6 self-assigned addresses, because the probability of two nodes choosing the same address (a.k.a address collision) is given by the birthday paradox $p(X(n_t)) = 1 - \prod_{i=1}^{n_t-1}(1 - i/2^{16})$, in which X is a discrete uniform distribution of addresses and n_t is the total number of nodes deployed. As it can be seen in Figure 1, the collision probability is over 10% with only 120 nodes and reaches 50% with ~300 nodes.

The 802.15.4 link layer address assignment protocol (LLAA) assumes the existence of a single PAN coordinator. The coordinator is responsible for defining the address domain to be used with the PAN and for dividing that address domain between its direct neighbors. Each of the neighbors takes one address for itself and divides the remaining addresses by its direct neighbors. The protocol continues until every node has an address. This protocol clearly does not scale well; it is prone to waste of address space if the tree created during the assignment is not balanced; it does not handle well node hording; and it is easily attacked by a malicious node near the coordinator; for example, it may take every address for itself, thus preventing others from getting addresses.

Besides LLAA, several other proposals have been made to dynamically assign short unique addresses to nodes [7–14]; however, most assume that every node in the PAN must have a unique address [7–9, 11] and/or that there is one central coordinator for each PAN [10, 12–14]. While the latter assumption is mostly true, although not always, the former requirement is often unnecessary.

Addresses are necessary both for identification and routing; however, both global identification and routing are often done with other types of identifiers. Often, the only requirement is that the addresses are locally unique, that is, that only the direct neighbors of each node have distinct addresses [15, 16].

In WSNs, nodes are often identified by data attributes and not by their unique global addresses. For instance, in the direct diffusion protocol [17], communication is data centric. A node requests data by sending an interest to named data. This interest is propagated to its neighbors, building an interest tree. Whenever a source needs to send data, the data flows hop by hop over that interest tree to all nodes that have manifest interest in it. In this situation, there is no need for a global node identification, since only data must be named, although local addresses are still needed to exchange messages between neighbors.

On the other hand, 6LowPan drafts define two types of routing: "mesh-under" and "route-over" routing [4]. The former usually requires PAN unique addresses, unless some sort of direct diffusion routing protocol is used, but the latter only requires local-unique MAC addresses, because the actual routing is done with IPv6 addresses. In this case, the MAC addresses are used to distinguish the message source and destination within the same "network segment," that is, within the radio range of the source node, because every node behaves as an IPv6 router. There are several reasons to prefer "route-over" to "mesh-under" routing, the main one is that every existing network diagnostic tool for IP management will not work if a "mesh-under" routing is used [18].

Assuming that only local unique addresses are needed, the problem is much more simple, but we still cannot use random self-assigned addresses, because the probability that k nodes, in the direct vicinity of each other, choose the same address over an address space of size $|A|$, on a network with N nodes and an average of $\overline{n}$ neighbors per node, is given by (1)

$$P_{\text{col}}\left(k\text{-}way\right) = 1 - \left(1 - \left(\frac{1}{|A|}\right)^{k-1}\right)^{\left(\binom{\overline{n}+1}{k} + (N-\overline{n}-1)\times\binom{\overline{n}}{k-1}\right)}. \tag{1}$$

As it can be seen in Figure 2, the number of nodes that need to be deployed to have a 10% collision probability with an average of 20 neighbors, with an address space of 16 bit, is just 356.

1.2. Proposal. We have developed a message efficient and secure protocol to ensure the distribution of local unique

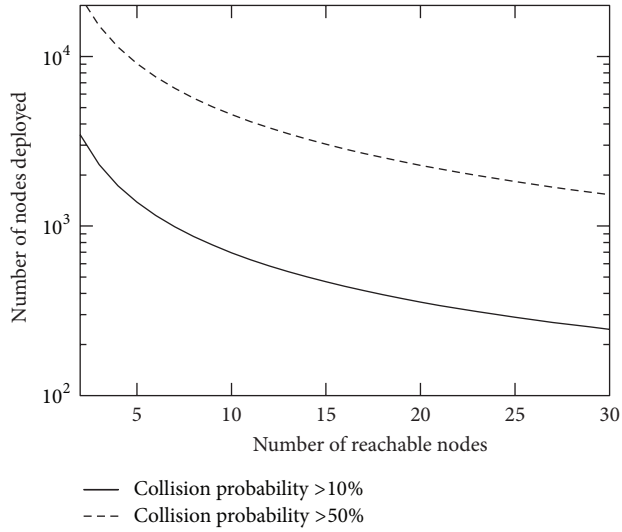


Figure 2: Number of nodes deployed for achieving a 10% and 50% collision probability with the average number of nodes directly reachable by each node.

addresses among neighbor nodes. The protocol is more efficient in terms of number of messages than similar ones, and it is secure against a bounded number of malicious nodes and is able to handle late deployment scenarios and partition rejoining without resetting established addresses (i.e, without breaking established routes between nodes).

The protocol efficiency is obtained through the use of only negative acknowledgements and through an improvement over a previously proposed flooding control technique. The security features of the protocol are obtained without cryptography, through a technique named *whispering*. We are assuming that sensor nodes are sold without key material in place. If cryptographic keys are going to be needed, they will be distributed later over the wireless medium together with sensor programming. Finally, the solution to handle partition rejoining is accomplished using *nicknames* between sensors.

The next section describes relevant related work. Sections 3 and 4 describe and evaluate the basic protocol and the solution to avoid intruders, respectively. Section 5 describes the solution to handle the incremental deployment of new sensors, and, finally, Section 6 concludes and presents future work.

2. Related Work

The naming problem which we intend to solve has been addressed before for WSNs and for mobile ad hoc networks (MANET). The IETF Zeroconf working group proposed a solution for MANETs [7] that rely on the discovering abilities of the underlying routing protocol. In this proposal, each node independently chooses an address and then sends a routing requesting packet to that address. If a route is found within a timeout period, it concludes that the address is already in use; otherwise, the address is not used and the protocol ends. The main problem of this protocol is the definition of the timeout when the number of hops needed to reach every node in the network increases. The protocol was developed for MANETs and does not scale well for WSNs where the number of nodes and hops between them is much higher.

Unlike the Zeroconf working group proposal, most naming solutions' goal is to find a unique 2-hop address. This problem is known as the neighborhood unique naming (NUN) problem and is similar to the classical coloring graph problem with conditions at distance 2. In [19], it is proven that *there is no determinist self-stabilizing algorithm to solve the NUN problem in uniform and anonymous networks under a distributed scheduler,* and it proposes a self-stabilizing probabilistic algorithm. The algorithm is very simple: each node keeps two variables, one with its address and one with the address of two colliding nodes in its neighborhood; if there are no collisions in the neighborhood, the second variable is empty; each node starts by asking every neighbor their address to calculate the second variable; it then asks its neighbors for their variables' values; if any of these values is equal to its own address, the node randomly chooses another. The algorithm was proven to self-stabilize, although no protocol was given to implement it. In particular, it is not clear how messages from two distinct nodes with the same address can not be confused with a repetition of a previous message.

The same strategy is followed in [20], but instead of using the 2-hop neighborhood, it uses a 3-hop neighborhood and a cache in every node to keep the addresses of its 3-hop neighborhood. It claim, that by using the 3-hop neighborhood it bounds each node number of attempts to choose an address. However, no consistency protocol for the 3-hop neighborhood cache is given, which makes it difficult to calculate the average number of messages required to reach a consistent state.

A cache is also used in [15] to keep addresses of direct neighbors. In this proposal, each node sends a periodical message with its address. This message is stored in the cache of its neighbors. If a node detects that two of its neighbors have the same address, it sends a warning message to one of them. With this protocol, nodes may change addresses several times during the lifetime of the network which may not be acceptable by every application or routing protocol. Moreover, the periodic broadcasting of addresses may be too energy expensive, and the authors fail to prove the self-stabilization of the protocol.

The approach followed in [8, 9] is different but also probabilistic. They leverage on the wireless nodes' ability to detect media access collisions to know if there are other nodes contending for an address or not. If a node discovers that no one else is broadcasting at the same time, it takes the address for itself, and everyone else knows that that address is taken. If several nodes broadcast at the same time, they all flip a coin to decide if they will participate in the next round. On average, only half of the contenders transmit in the next round. After several rounds, only one node will transmit and will get the address. Although simple, this solution assumes that the radio is able to listen at the same time it transmits, which is not true in most inexpensive radio transmitters.

The solution proposed in [15] is able to handle dynamic scenarios by periodically repeating the protocol. In this proposal, each node sends a periodical message with its address. Each node keeps a log with every message seen by it; if it detects a duplicate address, it sends a warning message to both nodes. Upon receiving the warning message, both nodes choose another address and announce it again. With this protocol, nodes may change address several times during the life-time of the network which may not be acceptable by every application or routing protocol. Moreover, the periodic broadcasting of IDs may be too energy expensive, and it is not clear how the periodic address announcement is not mistaken with an address collision.

The work of [11] proposes the use of the location of each node in space to assign a unique field address but does not describe how to run a localization protocol without addresses or how to cope with common localization errors produced by such protocols.

With the exception of the solution described in [21], most 2-distance graph coloring algorithms and address assignment protocols are either deterministic and semicentralized or distributed and probabilistic. The reason why distributed protocols are probabilistic is the fact that, under a distributed and unfair scheduler, every node may precisely copy all the other nodes' movements always choosing the same addresses; thus, the algorithm may never end. Clearly, a deterministic solution is better than a probabilistic one, because there is always the possibility that it never ends. However, most deterministic solutions do not scale well because they are either centralized or semi-centralized.

The centralized solution is never used in MANETs or WSNs. It would be similar to having a DHCP server replying to every node, which clearly does not scale beyond a few dozens of nodes. The semi-centralized solutions work by starting the assignment process at one specific central node and then distributing the assignment workload among other nodes. The DRCP and DAAP protocols [10] work together to assign addresses in MANETs. The DRCP is used by the node requesting an address as in DHCP: the node starts by asking if any of its neighbors is acting as a DRCP server, and if some of them reply, it chooses one of them to get the address from. After having received the address, the node uses the DAAP protocol to ask its DRCP server for half of its pool of addresses, and it then proceeds by acting as a DRCP server.

As described before, the 802.15.4 protocol uses two types of addresses: 64 bit global unique addresses and 16 bit network unique short addresses. The 64 bit addresses are used at the beginning of the network deployment to establish the 16-bit addresses, which are used thereafter. The protocol which establishes the 16-bit addresses is similar to DRCP/DAAP. However, instead of using two distinct steps for assigning an address and for assigning a pool of addresses, 802.15.4 only uses one; instead of giving up half of its address space to each child node, a node equally divides its pool of addresses among its neighbors.

The 802.15.4 distribution of addresses is specially unfitted for very unbalanced field topologies. This problem is handled in [12] by adapting the address distribution strategy to the network topology, at the cost of some more messages. The work of [13] also tackles the same problem but chooses a different solution. Although centralized, the proposed protocol does not define the addresses centrally; instead, it uses the relative hop-count location of each node to the other nodes to build each node address. The central node is used just as the center of a radial coordinate system, which is built by exchanging messages with that central node.

Another centralized protocol is proposed in [14]. Although the goal of the protocol is the establishment of a field unique address, it starts by specifying a local unique address to allow for point to point communication in the rest of the protocol. However, the description of this first phase is very short, and it advocates that a simpler beacon system is enough to detect address collisions within the protocol, which is not true because the address must be unique in a 2-hop scenario and not only in a 1-hop scenario.

None of the centralized or semi-centralized address assignment protocols scale well when the number of nodes is too large or the address space is too small. The solution presented in [21] does not have this problem, but it is costly in terms of time. In essence, the solution uses a token to establish an order between nodes' address assignment, which in a network of several hundred nodes may take some time. Moreover, this solution requires the synchronized update of state variables in both sender and receiver nodes. This is a problem when nodes do not have a valid address, because in such situations, it is difficult to establish a single receiver.

Finally, most 2-distance graph coloring algorithms [21–23] try to find the graph coloring which uses the minimum number of colors. We have a much more relaxed goal. We want to find a 2-distance graph coloring with a minimum number of messages, bounded by a maximum of 2^{16} colors.

3. Basic Protocol

The basic protocol objective is twofold: (i) ensure a unique local identification on the WSN over a distance 2 neighborhood with an arbitrary large probability $p < 1$ and (ii) minimize the energy loss by minimizing the number of messages sent and received.

The protocol assumes no local or global knowledge of topological information. This includes global and relative geographic coordinates, number of neighbors, local and global density, or even the global number of nodes. This is important in a scenario where most sensor nodes do not have a GPS module and are distributed randomly over the sensor field. In such scenario, it is not possible to know geographic information at every sensor without running a localization protocol, which can only be run if proper addressing is in place. Therefore, although topological information may be acquired in the future, it is not available at initialization time.

Unlike several initialization protocols [19, 20], we have chosen to keep state variables private to each node; that is, we avoid the use of caches with partial knowledge of the state variables of other nodes. Although such caches would improve the nodes' knowledge over their neighborhood, we avoid expensive cache coherence protocols.

The basic protocol is very simple, each node chooses a random address for itself and asks its neighbors if they have chosen the same address. If at least one of them has chosen the same address, it replies with a NACK, saying that a collision was found; otherwise, each receiving node rebroadcasts the query to its own neighborhood. The nodes receiving these rebroadcasts check the receiving packet for a collision with their own addresses. If they find a collision, they reply in the same way as the first hop nodes does; otherwise, they do nothing. A complete description of the protocol is given in Listings 1, 2 and 3. Listing 1 contains the main functions, the `init()` function initiates the address assignment protocol of each node, and the `receive()` function handles the reception of every message from the protocol.

Notice that there are no positive replies, only negative ones. This is because the probability of finding a node with the same address in a 2-hop neighborhood is very low; thus, in the usual scenario, only query messages are sent. Note that the probability that a node in a 2-hop neighborhood chooses the same address of the query node is given by $p_c(|A|, \overline{n}) = 1 - (1 - 1/|A|)^{4\overline{n}}$, that is, $\sim 10^{-3}$ for a 20 node neighborhood and a 16-bit address space, which is much lower than the probability of finding a collision in a 2-hop neighborhood (the birthday paradox).

The first problem that the protocol needs to overcome is how to distinguish the rebroadcast messages originated in itself from the ones originated in other sensors. If a node trying to establish an address receives a query for that same address, it should answer declaring that that address has been taken, even if that action results in neither of the nodes sticking with the address. However, if the node is hearing an echo of its own query, it should do nothing. Thus, we need a way to uniquely identify the messages.

The messages sent by each node are stamped with a collision free 64 bit node address (extended address). This extended address can be a manufacturer unique number, when available, or a random number generated whenever a node starts. However, as we will describe later, a random number is preferred over a manufacturer unique number for security reasons. Note that extended addresses are only used in the context of the initialization protocol. Afterwards, only short addresses are used. In fact, the protocol can be seen as a recoloring protocol with a smaller color space.

Two other similar problems happen when a rebroadcast node needs to relay a NACK back to the original querying node and when a node receives a NACK for its own address. In both cases, nodes should only act upon NACKs which were triggered by their own queries; otherwise, the protocol may not stabilize.

Self-stabilization, as defined in [24], is an important property of a distributed protocol. It ensures that regardless of the initial state of the system and regardless of the scheduling of actions taken by each participating node the system will reach a legitimate final state in a finite number of steps.

Beauquier et al. [25] redefined self-stabilization for probabilistic protocols in such a way that regardless of the initial state of the system and regardless of the scheduler strategy the system will reach a legitimate final state with probability 1. Using the framework for proving self-stabilization of probabilistic protocols defined in [25], it can be shown that the previously described protocol, satisfies the above mentioned definition of self-stabilization, if extended addresses are used to link queries and replies.

Informally, the framework, defined in [25], states that a probabilistic protocol is self-stabilizing for a given specification if there is a sequence of predicates over system states $L_i(S_k) \cdots L_n(S_k)$ where S_k is the system state at step $k \geq 0$ and $n > i \geq 0$ such that the following conditions hold.

(i) The last predicate $L_n(S_k)$ (known as the legitimate predicate) of the sequence is a predicate that identifies a legitimate final state according to the specification.

(ii) For every scheduler, the probability of reaching a system state satisfying the specification from a state verifying the legitimate predicate is 1, which can be formalized by the following conditional probability:

$$P\left(\frac{L_n(S_{m+k})}{L_n(S_m)}\right) = 1, \quad k > 0,\ m \geq 0. \tag{2}$$

(iii) For every scheduler strategy, if the probability of reaching a state verifying one predicate in the sequence is 1, then the probability of reaching a state verifying the next predicate in the sequence is also 1, which can be formalized by the following conditional probability.

$$P\left(\frac{L_{i+1}(S_{m+k})}{L_i(S_m)}\right) = 1, \quad k \geq 1. \tag{3}$$

The first two are easily verified by the protocol. If we choose n to be the total number of nodes, and $L_i(S_k) = \{N_{\text{cf}}(S_k) \geqslant i\}$, where $N_{\text{cf}}(S_k)$ is the number of nodes with a collision free address in state S_k, the last predicate ($L_n(S_k)$) clearly identifies a legitimate final state (first requirement). Moreover, after reaching a legitimate state (i.e. every node has a 2-hop unique identifier), the protocol ceases to send NACKs. Since addresses are only changed when a NACK arrives, the system will reach a final configuration verifying the specification (second requirement).

To prove that the protocol satisfies the last requirement, we will use another result from [25]. It states that the third requisite is verified if predicates are *closed* and verify the local convergence property. A predicate L_i is closed if for every S_k and $m > 0$, $L_i(S_k) \Rightarrow L_i(S_{k+m})$. Two predicates are said to verify the local convergence property if, according to a scheduling strategy, the probability of reaching a state verifying the second predicate from a state verifying the first predicate, in less than $k > 1$ steps, is greater than $\delta > 0$.

The predicate $L_i(S_k) = \{N_{\text{cf}}(S_k) \geqslant i\}$ is closed under the given protocol, because whenever a node chooses a 2-hop unique address and every 2-hop neighbor has the opportunity to reply and does not do it, the node sticks with that address forever, provided that it only acts upon NACKs to its own queries. Notice that, if the node tries to optimize the process of detecting a collision by overhearing NACKs

```
typedef enum {
 Query, NACK, ColQuery, ColReply, ColSolve} msgtype_t;          // Message Types.
typedef struct {                                                // Message structure.
 uint16_t d_add, s_add;                                         // Destination and source addresses.
 msgtype_t type;                                                // Message type.
 byte      hop;                                                 // First or second hop.
 uint64_t xadd[4];                                              // Vector with extend addresses.
} msg_t;
uint16_t myAdd;                                    // Assigned short address
uint16_t queryPower, replyPower;                   // Query and Reply transmission power variables.
uint16_t pwrStep                                   // Power step reduction.
uint64_t myXAdd, prev_xadd;                        // Extended address and previous extended address.
Name: init
Description: Starts the address self-assignment procedure.
init() {
 myXAdd= largerandom();                                         // Choose a random extend address
 queryPower = replyPower= MaxPower;                             // Sets the power step
 pwrStep = (MaxPower − MinPower)/NCircles;                      // for each ring
 newAddress();                                                  // Chooses a new address
}
Name: receiveMsg
Description: Processes each received message. The "Query" and "NACK" message types are part of the address
             assignment protocol, while the "ColQuery"and "ColReply" are part of the collision solving protocol.
void receiveMsg(message_t msg) {
 switch(msg.type) {
  case Query:                                   // Query message received
    if(msg.s_add== myAdd && msg.xadd[0]!= myXAdd) {// Test if there is a collision
      sendNack(msg);                            // Send a NACK if there is a collision
    } else if(!duplicate(msg)){                 // Test if a copy was previously received.
       if( msg.hop == 0)  {                        // If it is a fist hop query schedule
          msg.hop = 1;                          // a message for transmission after some time.
          sendQueryAfter(msg, delayStep∗msg.strength/pwrStep)
       }
    } else if(incCtr(msg) > MaxCtr)             // Inc. and test the n° of copies
          markAsTrans(msg);                     // If bigger than threshold, remove
    break;                                      // the message from the sending queue.
  case NACK:                                    // Negative ACK received
    if(msg.hop==1 &&                            // If it is a NACK to a 2nd hop query
      (msg.xadd[1]== myXAdd || msg.xadd[1]== prev_xadd)) {  //  that was sent by me
      msg.hop = 0; sendNack(msg);               // Then send a NACK to the original query node.
    }
    if(msg.xadd[msg.hop]== myXAdd)              // If is is a NACK to a query sent by me
      if(add2Ctr(msg.hop) ||                    // Inc. the counter of NACKs and if exceeds the
        (msg.hop==0 && msg.s_id== myAdd))       // threshold or I'm the original query node (1st hop),
          newAdd();                             // choose a different address.
    break;
  case ColQuery:                                  // Receive a Collision Query message
    msg ={msg.s_add, myAdd, ColReply, myXAdd};    // Send a reply
    send(msg, replyPower);                        // with my short and extended addresses.
    break;
  case ColReply:                                  // Receive a Collision Reply message
    if(isNotMsgScheduledTo(msg.s_add, msg.xadd[0])) {       // if is the first message
      msg = {msg.s_add, myadd, ColSolve, msg.xadd[0]};      // Schedule a ColSolve msg
      scheduleMsgToSend(msg, timeout);            // with only its address
      falsePositives++;                           // If no more ColReplies arrive this is a false positive
    } else {                                      // If is the second message it confirms the collision
      addXAddToMsg(msg.s_add, msg.xadd[0]);  // Add the extended address to the msg
      markAdd(msg.s_add);                    // Mark the address as a collision
      falsePositives−−;                      // Confirm that it was not a false positive
    }
    break;}}
```

Listing 1: Main protocol functions. The `init()` and `receive()` functions are the main protocol functions.

```
typedef struct {                    // Node signal strength record
  uint_16 add;                      // the address of this record
  byte ss_avg[2];                   // signal strength average.
  byte n[2];                        // messages received.
  boolean solving;                  // is already solving a collision
} NodeRecord;
const int maxMsgCount = 15;

Name: recMsgPwr
Description: Record the message strength for each source address and type.
             Analyses the previous maxMsgCount measurements and decide to start a collision solving procedure.

void recMsgPwr(int add,             // Source address
     int isB,                       // 1-broadcast, 0-otherwise.
     int ss) {                      // signal strength.
   NodeRecord nrec = getAddRecord(add);          // Find the record of the node with that address
   if (nrec.solving )                            // If its already solving a collision for the node
     return;                                     // with that address, returns.
   if(nrec.n[isB] >)0 {                          // If the number of messages record for that address is >0
     int ci = 4 + falsePositives;                // Set the confidence level to 4 (4∗sigma: −99.99% ci) plus
                                                 // the number of false positives to dynamically adjust the ci.
     byte diff = abs(ss − nrec.ss_avg[isB]);     // Diff between the signal strength (ss) and the average ss
     if(diff > ci∗0.003∗ss)                      // If diff is bigger then the confidence level
         startColSolving(add);                   // starts the collision solving protocol
   }
   if(nrec.n[isB] > maxMsgCount) {               // If the number of record msgs is bigger then the maximum
     nrec.n[isB] = 1;                            // resets the count and
     nrec.ss_avg[isB] = ss;                      // the signal strength average
   } else {
     nrec.n[isB]++;                              // Increments the message record count
     int delta = ss − nrec.ss_avg[isB];          // calculates the new signal strength
     nrec.ss_avg[isB] += delta/nrec.n[isB];      // average.
   }
}
```

Listing 2: Detecting an address collision using previous information on signal strength.

to queries initiated by other nodes, it does not verify this property and may not stabilize.

Given the collision probability p_c and the probability of finding a 2-hop unique identifier $p_s = 1 - p_c$, the probability of collision after k independent trials is $P_c(k) = p_c^k$, and the probability of success is $P_s(k) = 1 - p_c^k$. Thus, if we take $\delta = P_s(k-1)$, then $P_s(k) > \delta$, provided that $p_c < 1$. Notice that after only three trials the collision probability is in the order of magnitude of $\sim 10^{-10}$ for networks with a neighborhood density from 8 to 16 nodes and a 16 bit address space.

3.1. Broadcast Problems. One of the previously described problems of the protocol is that it relies on broadcast messages. Broadcast messages are inherently unreliable because whenever the number of nodes in the neighborhood is not known, the emitter will not be able to know if messages have arrived or not. However, in WSNs, the problem is even worse because messages may not arrive for many more reasons than in other network scenarios:

(i) the well known hidden terminal problem in radio networks may prevent messages from arriving without being noticed by the emitter;

(ii) depending on the MAC protocol, nodes may have the receiver asleep, to prevent energy loss, when a broadcast message arrives.

The common solution to improve broadcast reliability is to repeat each broadcast message several times to improve the probability of being received. However, this solution increases the potential of message collision whenever several nodes are trying to broadcast a message. When some of these messages are rebroadcasts of previously arrived broadcast messages, we may be faced with the so called broadcast storm problem [26].

To reduce the broadcast storm problem, we use the counter-based solution proposed in [26] enriched with distance information. In the original counter-based solution, some nodes are prevented from rebroadcasting a received message in order to minimize the number of messages sent. Whenever a node receives several replicas of the same message, it concludes that most of its neighbors have already received the message; thus, it does not need to send it again. By avoiding sending messages, nodes are minimizing the broadcast storm problem and are saving energy, but they are increasing the probability of not reaching nodes that they should. In [26], it is shown that, in a homogenous

```
Name: newAddress
Description: Generate a new short address and send it after a random delay

void newAddress() {
  myAdd = random();                          // Generate a new random short ID.
  sendQueryAfter(msg, randDelay);            // Sends a Query for that short ID.
}

Name: add2Counter
Description: Counts 1st and 2nd hop NACKs and reduces transmission power accordingly
boolean add2Counter(byte hop) {              // Count the number of NACKs received.
  static int secCtr
  secCtr += hop==0?1:3;                      // A NACK at the 2nd hop decreases power faster than at 1st.
  if (secCtr>9 && queryPower>0) {            // If to many NACKs were received:
    queryPower -= pwrStep;                   // decreases the power, unless power
    secCtr = 0;                              // is already zero; resets the counter;
    prev_xadd = myXAdd;                      // saves the previous extended address;
    myXAdd= largerandom();                   // and generates a new one.
    return true;                             // Inform that a new short address must
  }                                          // be generated.
  return false;                              // There is no need to generate
}                                            // a new short address yet.

void sendQueryAfter(msg_t msg, int delay) {  // Query messages are sent
  msg.type = Query;
  msg.xadd[msg.hop] = myXAdd;                // with the extended addresses
  sendAt(msg, queryPower, time()+delay);     // and with current queryPower.
}
void sendNack(msg_t msg) {
  msg.type = NACK; send(msg, replyPower);    // NACK messages are sent
}                                            // with replypower.
```

Listing 3: Auxiliar functions.

radio network, the uncovered area of a rebroadcast is directly related to the number of copies already received. In the original implementation, nodes rebroadcast after a random delay, provided that in the meantime they have not received enough copies of the same message. In the proposed solution, nodes further away from the source broadcast first, thus increasing the probability that nodes closer to the source are prevented from broadcasting.

There are other methods to minimize broadcast storms with better efficiency ratios, that is, the ratio; between the covered area and the number of broadcasting nodes is better with other methods. However, all these methods require either the knowledge of the topological localization of each node [26] or, at least, each node's neighbors [27].

In the proposed protocol, after receiving a query message, the node checks if that message has been previously received. If the message has been previously received more than a specified number of times, the message is marked as transmitted. Otherwise the message is scheduled for broadcast after a delay directly proportional to the power of the received message (line 44 in Listing 1). The result is that the retransmission area is divided into concentric rings. The nodes in each of these rings rebroadcast at more or less the same time. Notice that rings are not evenly distributed in space because the reception power varies with the inverse square of the radius, which is more or less consistent with the error in measuring message strength, which is much bigger for low power receptions; that is, outer rings are wider than inner rings because outer nodes have less accurate positioning than inner nodes.

The first question that arises is the number of copies that need to be received in order to prevent the message to be rebroadcasted. Williams and Camp [28] found that for networks with densities lower than 11 neighbors this threshold must be ≥ 4 to get a maximum coverage, that is, minimize the number of nodes that never receive the message. However, their scenario is different from ours (we need to cover a 2-hop region while they need to cover the whole network), and they do not use the reception signal strength to schedule rebroadcasts.

We have modified the 802.11 MAC layer of the J-Sim simulator [29] to incorporate our identity assignment features and tested over a field of $n = 300$ nodes using the free propagation model. The sensors were placed randomly on a rectangular field with an area $A = (n/(k+1))\pi r^2$, where r is

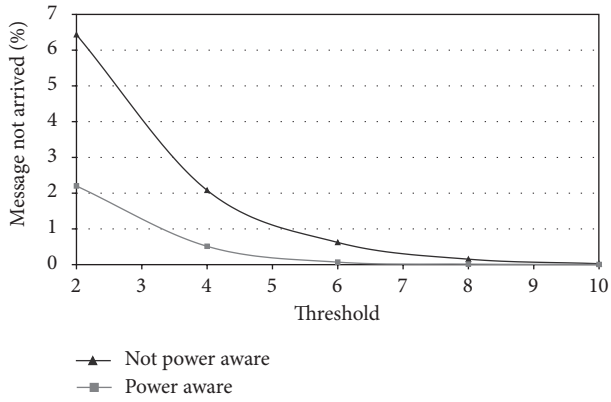


Figure 3: Impact of the threshold value on the percentage of messages not delivered, with and without power aware rebroadcast delay.

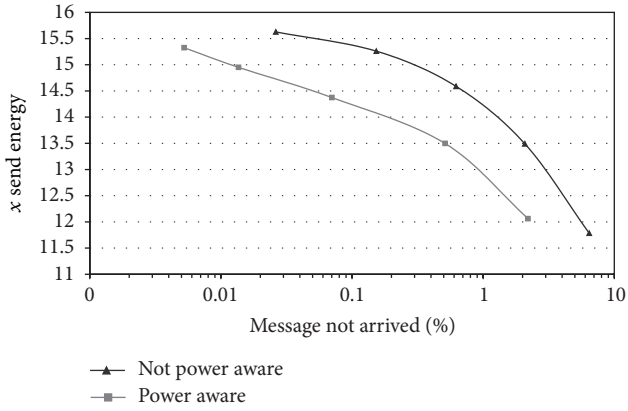


Figure 4: Energy spent by each node (divided by the energy spent by a single message transmission) for a given coverage.

the maximum transmission distance and $k = 12$ is the average number of neighbors of a node. The results are depicted in Figures 3, 4, 5, 6, and 7 and are analysed below.

The graph in Figure 3 shows the impact on the percentage of uncovered area with the chosen threshold. As expected, the uncovered area decreases when the threshold increases. However, it can be seen that the threshold required to achieve a significant coverage is much lower with the signal strength information than without it. To get a coverage of 99.5% (i.e., 0.5% of messages not received), we need a threshold of 6 without reception power information and a threshold of 4 with reception power information.

A lower threshold is better because it reduces the number of messages sent, thus improving energy consumption and minimizing the broadcast storm problem. In the end, the choice is between energy and coverage. Figure 4 shows the energy spent by each node as a function of the desired coverage. In this graph, we have assumed a simplified energy model in which sending a message consumes one energy unit, the reception of a message consumes 1/10 of a unit, and everything else is negligible.

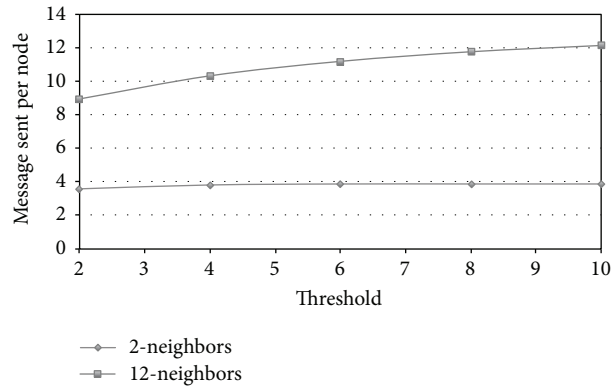


Figure 5: Number of messages sent by node with threshold value and density.

Again, as expected, the coverage increases with energy consumption both in the original solution and in the improved one. However, the solution which makes use of reception strength information is able to achieve better coverage with the same energy. In some cases, the uncovered area is 10 times smaller with the same energy consumption.

The resulting protocol is very fast. Figure 5 shows that each node sends around 4 to 12 messages on average to ensure the completeness of the protocol depending on the density of the field (threshold values have very little effect).

However, it is clear that even with this model some of the messages are not going to be delivered which may affect the correctness and stability of the protocol. It is obvious that the stability of the protocol is not affected because the number of NACK messages does not increase. On the other hand, the correctness is clearly affected because if a query or a NACK message does not reach some of the neighbors, two or more nodes may choose the same address without noticing. However, the probability of an undetected collision is very small (for a 99.5% coverage, the probability of a 2-*way* undetected collision on a network with 1000 nodes and 20 neighbors is ~0.1%) and may be handled by the protocol described in Section 5.

4. Avoiding Intruders

The previous scenario assumes that every node behaves well. If one or several nodes start replying to every query saying that they have already chosen that address, the well behaved nodes may end up with a depleted battery after repeating the query several times. If well behaved nodes do not share individual cryptographic key material with every neighbor, they are not able to distinguish well behaved neighbors from badly behaved neighbors. In such scenario, the only solution is to speak progressively softly until the badly behaved nodes are not able to hear the query. This is similar to whispering to your neighbor to prevent intruders from overhearing.

Whispering prevents nodes from communicating with more distant nodes which may have a negative impact on the network connectivity. We minimize this impact by reducing the power only as much as necessary and only in

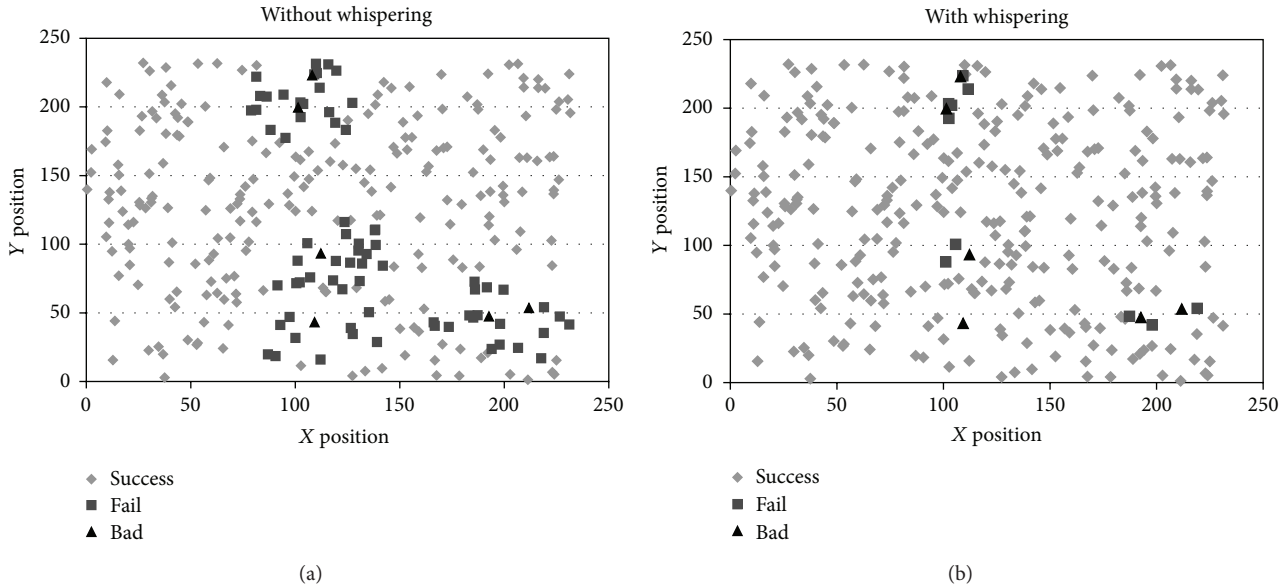


Figure 6: Impact of whispering over the percentage of affected nodes, in the presence of a percentage of badly behaved ones.

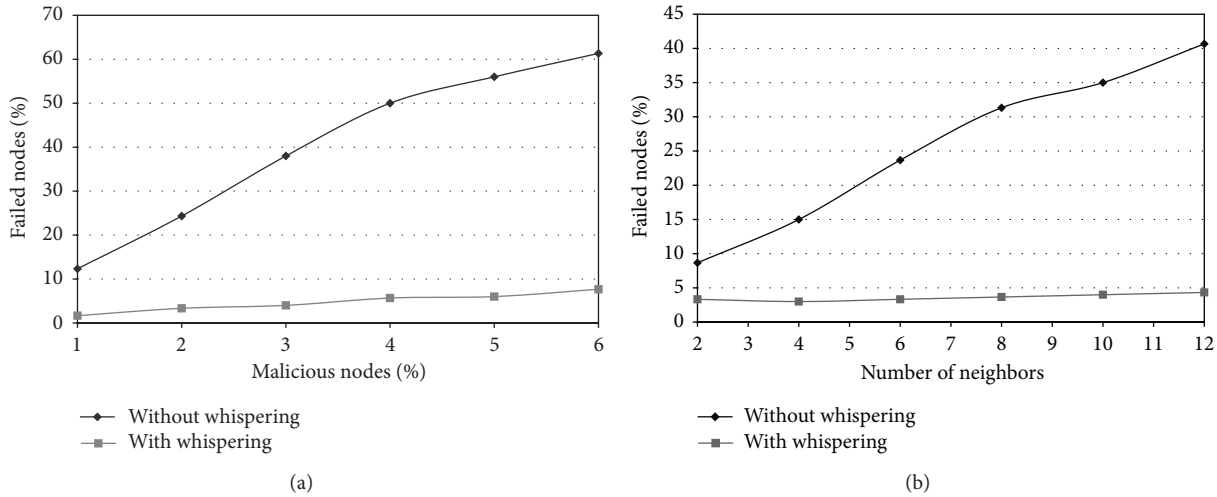


Figure 7: Relation between the percentage of failed nodes with the percentage of malicious nodes and network density, with and without whispering.

the nodes which are direct neighbors of the badly behaved one. Notice that, if a node receives a NACK originated at his neighbor's neighbor, reducing the power may prevent it from communicating with legitimate neighbors which are closer to it than the badly behaved node. It is its neighbor that should reduce transmission power. However, a node can not know for sure if a NACK is being relayed or produced at its neighbor, since a badly behaved node may always forge a NACK as if it were being relayed. Our solution was to reduce the power more quickly at nodes receiving NACKs to be relayed. Therefore, the only way a malicious node is able to force another node to reduce its transmission power rapidly is by being near; otherwise, it can only affect the node through relayed NACKs.

The reduction of transmission power should only affect queries, and the reply messages should be transmitted at full power; otherwise, a node could be prevented from sending a NACK only because it has a badly behaved node near it (see line 55 of Listing 1 and function `add2ctr()` in Listing 3).

After receiving a query from a node, a badly behaved node may start issuing NACKs to random addresses, even if it does not receive any more queries (because of query power reduction) trying to guess the next chosen address. To prevent it, a node should change its extended address every time it reduces its query transmission power.

A final word about the necessity of keeping the previous extended address after changing to a new one: the previous extended address is required whenever a node changes its address and it was already participating in another query as a relay node. If a NACK arrives, it must be relayed because there is no way to tell if that is a legitimate NACK from a colliding node or a malicious one.

This protocol is not able to completely prevent badly behaved nodes from stopping well behaved ones from choosing an address, but it minimizes the number of affected nodes. We have tested it by modifying a small percentage of nodes of our J-SIM simulator such that they behaved as malicious nodes would, if they wanted to prevent the protocol from succeed. We assumed that malicious nodes are also energy constrained and are not able to be radiating messages all the time. Instead, they reply with NACK messages to every query and continue to do so for a period after receiving the query, trying to guess the extended address used by the request that is whispering. As before, the number of nodes was set to 300 and the wireless range and deployment field size was chosen such that the average number of neighbors of each node is $k = 12$.

Figure 6 shows the effect of a small percentage of malicious nodes (2%) over a field of 300 randomly deployed nodes. Dark triangles represent malicious nodes, light rhombus represent nodes that were able to choose a collision free address, and dark squares represent nodes that were not able to choose an address or became isolated from nonmalicious nodes by the effect of power reduction.

As expected, the number of nodes which were not able to get an address with whispering is much smaller than without it. With whispering, the affected nodes are in the direct vicinity of the malicious nodes, while without whispering, the affected nodes are spread over their 2-hop neighborhood.

The number of affected nodes is obviously dependent on the number of badly behaved ones, but it is also dependent on the network density. The number of failed nodes increases when the number of nodes in the vicinity of malicious ones increases. Figure 7 shows how the percentage of affected nodes increases with the percentage of malicious ones and with the network density. In both cases, the percentage of failed nodes is much lower and increases much slower with whispering than without whispering. In fact, with whispering, the variation of failed nodes with the network density is almost negligible, while without whispering, the effect is very noticeable.

5. Handling Incrementally Deployed Scenarios

One important feature of address assignment protocols, which is often forgotten, is its ability to handle late deployed sensors and merging of network partitions. The deployment of additional sensors may be necessary either to improve the sensor coverage or to improve the network lifetime; the sensors in place may be at the end of its battery. The merging of network partitions may happen either because there was an obstacle dividing nodes at the time of deployment which is now removed or because the addition of new nodes made two or more networks reachable to each other.

In such scenarios, address collisions may happen, because at the time of address assignment, not every node knew about each other. Most address assignment protocols do not consider these scenarios, and the ones that do choose to rerun the assignment protocol in the colliding nodes [15]. This strategy may have a negative impact on routing, because every route established through those nodes needs to be rebuilt.

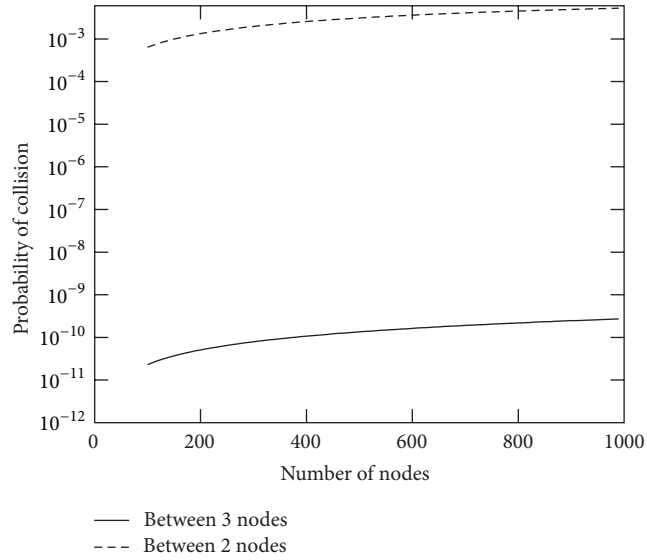


Figure 8: Collision probability between 2 nodes and between 3 nodes for a field with a density of 25 neighbors.

Another problem that these protocols need to handle is how to detect the existence of colliding addresses. In [15], address collisions are detected during the periodically neighborhood query which is done for this purpose. However, given that the addition of new nodes and the merging of networks are rare, such a scheme is too energy expensive. In [30] (a protocol designed for MANETs), each packet has an additional 64 bit unique number which is used to detected address collisions, but that is not an option in WSNs given the size of each packet.

Whenever a node is added or a barrier is lifted between two or more network partitions, it is possible that two or more nodes with the same address became reachable by a single node. A *k-way* collision happens whenever k nodes with the same address are reachable by one node, a.k.a the detecter. The probability of having a *k-way* collision after the address assignment protocol runs is given by the probability of having a *k-way* collision $P_{\text{col}}(k\text{-way})$ (1) times the probability of a *k-way* cut during the address assignment protocol. Assuming that a 2-*way* cut is given by $p_{\text{cut}}(2\text{-way})$, then the probability of a *k-way* cut is given by

$$p_{\text{cut}}(k\text{-way}) = \left(p_{\text{cut}}(2\text{-way})\right)^{\binom{k}{2}}, \tag{4}$$

because it requires a cut between every two pairs of nodes in the *k-way* collision.

Figure 8 shows the probability of collision for 3-*way* and 2-*way* collisions for several number of nodes after running the address assignment protocol, giving an address space $|A| = 2^{15}$ (15 bit addresses, we will use the extra bit for collision solving), a neighborhood of 25 nodes, and a probability of a 2-*way* cut of 1. The collision probability is around 10^7 times below for 3-*way* collisions in comparison with 2-*way* collisions.

Given the above results, we make the hypothesis that *k-way* collisions with $k > 2$ are extremely unlikely, and we are going to focus our efforts in detecting and solving 2-*way* collisions.

5.1. Detecting Address Collisions. Our approach to detect address collisions is motivated by the way that people distinguish two voices in a crowd. If one of the voices is loud and the other is soft, then there are probably two persons talking. If the heard sentences do not make sense because they seem garbled, then it is possible that they are produced by more than one person. Neither of these heuristics gives precise information about the existence of colliding addresses, but they may be used as triggers for a collision solving protocol.

The former solution is independent on the transport protocol, while the latter is not. In order to detect out-of-order messages, the transport protocol must have the notion of order which is not the case for many transport protocols in WSNs; this is why we have chosen the former solution.

Given the hypothesis that only 2-*way* collisions may happen whenever the network changes, only two scenarios are possible:

(i) the address of the added node is the same of one of the nodes already in the network, and both are reachable by a third node (merge of partitions),

(ii) the address of two of the nodes in the network is the same, and they are reachable by the new node (node addition).

The first scenario is simpler than the second, although, as we will see, they will be handled the same way. If the nodes in the network knew each other, they are able to know the signal strength (SS) average and standard deviation of messages sent by each other. If one of the nodes detects a message with a SS much different from the usual, it may suspect an address collision, although, to be sure, it will have to run the collision solving protocol described in the next section. The second scenario is a bit more troubling because the colliding nodes are both new for the detecting node.

We have started by using an algorithm from Knuth [31] to incrementally calculate the average $\overline{\text{SS}}$ and standard deviation σ_{SS} of the SS without having to keep all samples, that is, the calculus is incremental. In order to get a four nines confidence level in the collision detection, we checked if the SS of each message was within four times the standard deviation of the average (5):

$$\left|\overline{\text{SS}} - \text{SS}_i\right| < 4\sigma_{\text{SS}}. \quad (5)$$

Otherwise; we would signal a collision. However, we have realized that the SS average varies over time due to battery drain and environment changes, leading to large standard deviations and making the system irresponsive to address collisions. Figure 9 shows a typical signal strength over time of messages received in a field of MicaZ Motes using the TinyOS 802.15.4 stack; the measurements were taken between two sensors 1 meter apart sending a message to each other, every second, for 150 seconds long. The first approach to solve this problem was to calculate the SS average and standard deviation using only the last few messages; however, that too proved ineffective because the standard deviation with too few messages lacked the necessary precision.

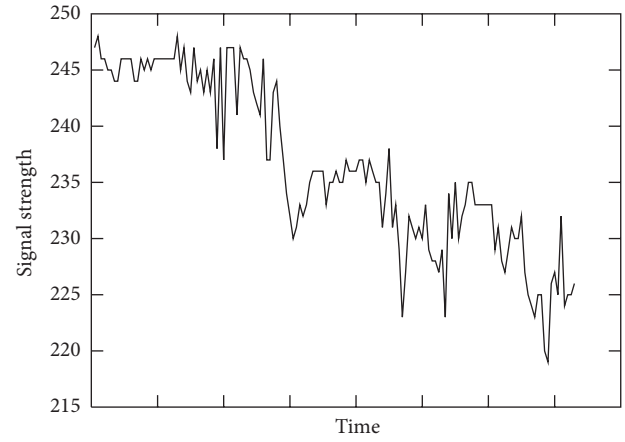


Figure 9: Signal strength of messages received by the same node over time.

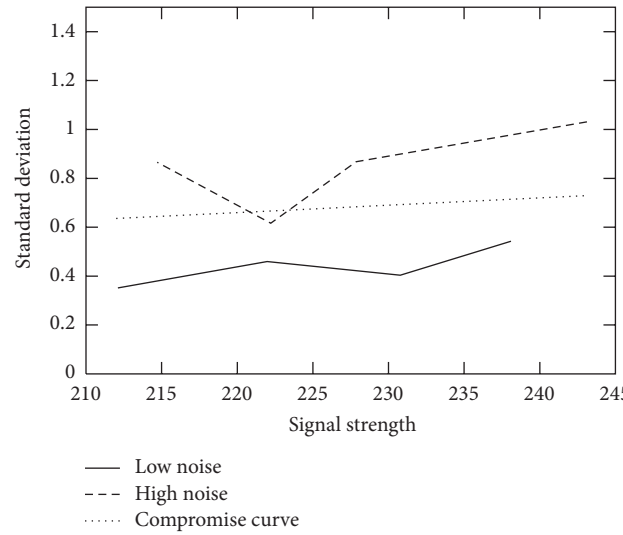


Figure 10: Standard deviation behavior with the signal strength ($\text{SS}_{\text{max}} = 255$) of messages.

Instead of computing both the average and standard deviation, we chose to compute the average over the last few messages and estimate the standard deviation based on the average. In order to do it, we have analyzed the standard deviation of messages' SS sent by the same node at different distances and in different places.

We have realized that the most relevant factor to the measurements of standard deviation is the place where the measurements were done, in particular if the test site has many wifi antennas working. This is consistent with several reports of interference between Wifi and 802.15.4 networks. We called such scenarios noisy ones and all the others nonnoisy since we did not find other relevant differences.

Figure 10 depicts the standard deviation variation with signal strength and noisy versus non-noisy environments. We

Table 1: Error rate at different distances (in meters) and noise levels.

False detections	Regulatory mechanism	Noisy environment					Nonnoisy
		1 m	5 m	10 m	15 m	Total	total
Negatives		0%	1%	3%	0%	**4%**	**0%**
Positives	Without	6%	1%	0%	4%	11%	0%
	With	0%	0%	0%	0%	**0%**	**0%**

have placed two nodes transmitting at 1, 5, 10, and 15 metters in noisy and non-noisy environments and measured the average and standard deviation of the received signal strength for each distance and environment over 150 messages. For each calculated average, we have plotted the correspondent standard deviation and drawn a line showing that, frequently, standard deviation is higher for higher reception strengths. As expected, the standard deviation over a few messages is very small in low noise scenarios and is almost twice in high noise scenarios (Figure 10).

A good compromise is given in Figure 10, in which the predicted standard deviation is given by $\widehat{\sigma}_{SS} = 0.003SS_i$. Our measurements showed that (5) with $\widehat{\sigma}_{SS}$ instead of σ_{SS} is suitable both for low and high noise scenarios; however, if for extremely high noise scenarios the number of false collisions detected by this protocol is too high, we incorporate a self-regulatory mechanism which increases the standard deviation every time a false positive is detected.

This solution has another advantage: it is able to handle both collision scenarios. Notice that, if a node arrives to a network and starts to communicate with two other nodes with the same address at the same time, it will not have previous information about average and standard deviation; therefore, it will not be able to find discrepancies with past history. However, with this solution, although the average will still be wrongly calculated because it will be something in between the two signals of the two communicating nodes, the standard deviation will not change much, which will allow the detection of the collision.

Listing 2 shows that whenever a message is received with a signal strength above or below a predefined confidence interval the solving protocol (see Section 5.2) is started. Notice that the set of values kept for each address is comprised by: the average of the signal strength of every message received (`ss_avg`); the number of messages received from that address; and a value indicating that a conflict is being solved. Notice, also, that the record structure uses two sets of values for each address, because it is expected that broadcast communications be done with a different transmission power than unicast communications. The transmission power of unicast communications is usually adapted to the distance between peer nodes, while the broadcast communications do not have this kind of adaptation.

We have conducted two sets of experiments, using MicaZ sensor nodes running TinyOS 2.0, in three different environments: two small non-noisy environments and one open and noisy environment, that is, a large student hall with many students moving, each one with its own laptop device with Wifi and many Wifi antennas in the vicinity. The first set of experiments was designed to detect false positives, and the second to detect false negatives.

5.1.1. False Positives. The false positive ratio is an important metric because it impacts the energy consumed by each node in the collision solving protocol; that is, running the collision solving protocol is expensive and should be triggered as seldom as possible. We have placed two nodes at different distances (1, 5, 10, and 15 meters) sending messages to each other and measured the amount of times that the algorithm detected a false collision (Table 1), that is, the number of times one of the nodes received two messages from a single node, with signal strengths different enough to be mistakenly identified as coming from two different nodes with the same ID. We have conducted 10 experiments at each distance and environment, each experiment exchanging 300 messages (150 messages each), and we have taken the average number of false positives. As expected, the number of *false positives* in the non-noisy scenarios was close to 0% which is consistent with the four nines accuracy specification. In the noisy environment, the system showed a rate of almost 11% of false positives without the regulatory mechanism, but with the regulatory mechanism, after just 3 false positives, it reached a steady state where no more false positives occurred (0% false positives).

Another interesting result is the distribution of false positives with the distance. Most false positives (6%) were experienced when the colliding nodes were 1 meter apart from the arriving node. For larger distances, within the transmission ratio of our nodes (in noisy environments 15 meters), the false positive rate is much lower, which may be related to the saturation of the signal (it is difficult to distinguish two persons shouting very near to us).

5.1.2. False Negatives. Although we have defined two collision scenarios: the arriving node connecting two networks previously disconnected and the arriving node having the same address of another node, we have only measured the false negatives in the former, because it is more general than the latter. The main difference between the two scenarios is the node that detects the collision; in the "connecting two networks scenario" it is the arriving node that detects the collision while in the "arriving node scenario with duplicate address" it is one of the other existing nodes that detects the collision. From a detection point of view the main difference between the two detecting nodes is that the former one has no history of messages received (it has just arrived) and the latter may have received messages before from one of the duplicate address nodes. Therefore the former scenario is

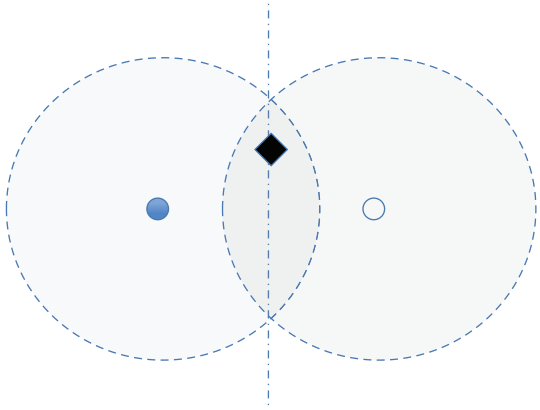


Figure 11: Area where the arriving node needs to be placed in order to detect the collision. The dark rhombus represents the arriving node and the light circles the colliding nodes.

harder to detect. Moreover, the algorithm specifies that after a predefined number of messages (`maxMsgCount`) the history of received messages is cleaned, therefore in the long run the two scenarios are equal.

In order to assess the false negative rate of the first scenario, we have chosen the worst possible placement of nodes for the purpose of collision detection. Assuming that all nodes have an equal radio range, the arriving one will be in the position of connecting two previously disconnected nodes with potentially the same address if it is placed within the intersection of the two radio coverages, and the difficulty of detecting the collision will be higher when the node is placed at the exact same distance of the two colliding nodes (Figure 11), since it will receive messages from both of them with similar strength. Again, we have conducted 10 experiments with 150 messages each at each combination of distance, environment, and regulatory mechanism (with and without) and taken the average number of times that the collision was not detected.

The protocol shows 0% of *false negatives* (Table 1) if it is closer to one of the colliding nodes or if it is run in a low noise environment; however, in a high noise environment with the two colliding nodes at exactly the same distance, we have experienced a false negative rate of 4%. A false negative does not mean that the arriving node will never detect the collision, it just means that it is not able to detect the collision within the frame period of 2 times `maxMsgCount`. Nonetheless, fast detection is important to minimize the impact of garbled communications.

5.2. Collision Solving Protocol. The collision detection protocol described in the previous section does not provide a definitive answer on the existence of an address collision. It ends by sending a query to every node with a specific address. Only if several nodes reply (with different extend addresses) the collision is confirmed.

When a node detects a collision, it gives to every node with the same address a nickname, and informs the node of that nickname The situation is similar to having two students in the same class named John, and we refer to one as "Little John" and to the other as "Big John." Notice that they will still be named John for every one else, and we cannot just name them "Little" and "Big," because we would create other collisions.

The solution is to reserve one bit from the 16 bit addresses for nicknames. Therefore, only 15 bit of the 16 bit addresses are assigned by the address assignment protocol, and the remaining bit is originally set to zero. When a node detects a collision it informs each of the colliding nodes that one of their addresses will have to set the bit to one. Each of the colliding nodes stores in a table the nickname for which it is known by that node. Whenever the node that changed its address receives a message from the node that detected the collision, it will only accept it if the bit on the destination address is set to one. Notice that other nodes continue to communicate with the node that changed the address with the old address; the change is only relevant for the communication with nodes that detect the collision.

This solution is only able to solve a single collision. If the address of a node collides with two other nodes, the protocol does not work because it would be possible for a node to end up being known by two nicknames by two different nodes, which would have a negative impact in broadcast communications. In fact, unicast communications would not be affected because the node could choose the nickname to use depending on the message destination; however, for broadcast communications, the node would not know the nickname to choose. Nevertheless, this is not a big problem because 3-*way* collisions are much less probable than 2-*way* collisions.

The proposed algorithm is shown in Listing 1. The algorithm assumes that the node detecting the collision (the initiator) has sent a "collision query" message (`ColQuery`) to all the nodes with the colliding address (line 26 in Listing 2). After receiving that message, a node replies with its extended address. When the initiator receives a "collision reply" (`ColReply`) message, it schedules a "collision solving" message (`ColSolve`) to be sent after a predefined timeout. If the node receives another "collision reply" message with a different extended address, it confirms the existence of a collision. If that happens, it marks the address that has a collision and modifies the "collision solving" message waiting to be sent by adding the new extended address. Finally, upon receiving the "collision solving" message, the colliding nodes choose independently the one that is going to adopt a nickname by comparing the extended addresses. The one with the smallest extended address adds a nickname to its address. Note that whenever there is no collision, that is, the collision detection protocol had a false positive none of the nodes adopts a nickname, because the smallest extended address in the message is 0 which is an invalid extended address.

The `cleanMsg` function is used whenever a message is received. If the message comes from someone with a nickname and the receiver has not detected that collision (some other node did), the nickname bit is clean. This ensures that a node that adopted a nickname may use it in broadcast communications.

6. Conclusion

The address self-assigning problem is a well-studied problem in the MANET world, but it has not received much attention in the WSN world. In this paper, we have described a simple address self-assignment protocol and proved its correctness. To improve the protocol performance, we have proposed an improvement to a well-known method of controlling message floods, based on the level of the power of message reception.

We have introduced the *whispering* technique to handle intruders when cryptographic keys are not available or have been compromised and show how to use it in the proposed protocol. We believe that this is a valid security technique and intend to study its application in other protocols.

We have designed and tested a very energy efficient mechanism (it does not use specific messages for that purpose) to detect late address collisions with a very low error rate. Finally, we have proposed the use of *aliases* to handle late address collisions without disrupting routing and other session tables.

The combination of all these protocols result in a very robust address assignment framework which was implemented in TinyOS 2.0 and tested in MicaZ motes.

Acknowledgment

This work was supported by national funds through FCT—Fundação para a Ciência e a Tecnologia, under project PEst-OE/EEI/LA0021/2013.

References

[1] P. J. Marrón, M. Gauger, A. Lachenmann, D. Minder, O. Saukh, and K. Rothermel, "FlexCup: a flexible and efficient code update mechanism for sensor networks," in *Wireless Sensor Networks*, vol. 3868 of *Lecture Notes in Computer Science*, pp. 212–227, 2006.

[2] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki—a lightweight and flexible operating system for tiny networked sensors," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04)*, pp. 455–462, November 2004.

[3] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals," RFC, 4919 (Informational), 2007.

[4] Z. Sheby, P. Thubert, J. Hui, S. Chakrabarti, and E. Nordmark, "LowPan neighbor discovery extensions," Internet-Draft draft-ietf-6lowpan-ipv6-nd-02, Internet Engineering Task Force, 2009.

[5] "IEEE.IEEE Std 802.15.4: wireless MAC and PHY specifications for LR-WPAN," IEEE Computer Society, 2003.

[6] L. A. N. Wireless, "Medium access control (MAC) and physical layer (PHY) specifications," IEEE Std, 802, 2007.

[7] A. Mobile, C. E. Perkins, and S. R. Das, "IP address autoconfiguration for ad hoc networks," Internet Draft draft-ietfmanet-autoconf-01.txt, Internet Engineering Task Force, MANET WG, 2000.

[8] K. Nakano and S. Olariu, "Randomized initialization protocols for ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 7, pp. 749–759, 2000.

[9] A. Micic and I. Stojmenovic, "A hybrid randomized initialization protcol for tdma in singlehop wireless networks," in *Proceedings of the International Parallel Distributed Processing Symposium (IPDPS '02)*, pp. 147–154, 2002.

[10] A. J. McAuley and K. Manousakis, "Self-configuring networks," in *Proceedings of the 21st Century Military Communications Conference (MILCOM '00)*, pp. 315–319, IEEE Computer Society, Los Angeles, Calif, USA, October 2000.

[11] Y. Tian, M. Sheng, and J. Li, "Virtual grid spatial reusing algorithm for MAC address assignment in wireless sensor network," in *Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA '06)*, vol. 1, pp. 649–654, Vienna, Austria, April 2006.

[12] M. S. Pan, H. W. Fang, Y. C. Liu, and Y. C. Tseng, "Address assignment and routing schemes for ZigBee-based long-thin wireless sensor networks," in *Proceedings of the Vehicular Technology Conference (VTC '08)*, pp. 173–177, Singapore, May 2008.

[13] Y. Liu and L. M. Ni, "Location-aware ID assignment in wireless sensor networks," in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Sysetems (MASS '06)*, pp. 525–529, Vancouver, Canada, October 2006.

[14] J. Lin, Y. Liu, and L. M. Ni, "SIDA: self-organized ID assignment in wireless sensor networks," in *Proceedings of the IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS '07)*, pp. 1–8, Pisa, Italy, October 2007.

[15] C. Schurgers, G. Kulkarni, and M. B. Srivastava, "Distributed assignment of encoded MAC addresses in sensor networks," in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '01)*, pp. 295–298, ACM Press, October 2001.

[16] C. Ribeiro, "Robust sensor self-initialization: whispering to avoid intruders," in *Proceedings of the International Conference on Emerging Security Information, Systems, and Technologies (SECURWARE '07)*, pp. 101–107, IEEE Computer Society, Valencia, Spain, October 2007.

[17] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, 2003.

[18] G. Mulligan, "The 6LoWPAN architecture," in *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets '07)*, pp. 78–82, ACM Press, New York, NY, USA, June 2007.

[19] M. Gradinariu and C. Johnen, "Self-stabilizing neighborhood unique naming under unfair scheduler," in *Euro-Par 2001 Parallel Processing*, vol. 2150 of *Lecture Notes in Computer Science*, pp. 458–465, 2001.

[20] T. Herman and S. Tixeuil, "A distributed TDMA slot assignment algorithm for wireless sensor networks," in *Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS)*, vol. 3121 of *Lecture Notes in Computer Science*, pp. 45–58, 2004.

[21] D. Angluin, J. Aspnes, M. J. Fischer, and H. Jiang, "Self-stabilizing population protocols," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 3, no. 4, article 13, 2008.

[22] M. Gairing, W. Goddard, S. T. Hedetniemi, P. Kristiansen, and A. A. McRae, "Distance-two information in self-stabilizing algorithms," *Parallel Processing Letters*, vol. 14, no. 3-4, pp. 387–398, 2004.

[23] T. Moscibroda and R. Wattenhofer, "Coloring unstructured radio networks," in *Proceedings of the 17th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '05)*, pp. 39–48, ACM Press, New York, NY, USA, July 2005.

[24] E. W. Dijkstra, "Self-stabilizing systems in spite of distributed control," *Communications of the ACM*, vol. 17, no. 11, pp. 643–644, 1974.

[25] J. Beauquier, M. Gradinariu, and C. Johnen, "Randomized self-stabilizing and space optimal leader election under arbitrary scheduler on rings," Tech. Rep. 99-1225, Universite Paris Sud, 1999.

[26] Y. C. Tseng, S. Y. Ni, Y. S. Chen, and J. P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Networks*, vol. 8, no. 2-3, pp. 153–167, 2002.

[27] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," in *Proceedings of the 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (ACM MSWiM '00)*, pp. 61–68, ACM Press, New York, NY, USA, August 2000.

[28] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '02)*, pp. 194–205, June 2002.

[29] A. Sobeih, J. C. Hou, L. C. Kung et al., "J-Sim: a simulation and emulation environment for wireless sensor networks," *IEEE Wireless Communications*, vol. 13, no. 4, pp. 104–119, 2006.

[30] N. H. Vaidya, "Weak duplicate address detection in mobile ad hoc networks," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '02)*, pp. 206–216, ACM Press, June 2002.

[31] D. E. Knuth, *The Art of Computer Programming: Seminumerical Algorithms*, vol. 2, 1981.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 806505, 13 pages
http://dx.doi.org/10.1155/2013/806505

Research Article

On the Security of Data Collection and Transmission from Wireless Sensor Networks in the Context of Internet of Things

Hong Yu,[1] Jingsha He,[2] Ruohong Liu,[3] and Dajie Ji[4]

[1] *College of Computer Science and Technology, Beijing University of Technology, Beijing 100124, China*
[2] *School of Software Engineering, Beijing University of Technology, Beijing 100124, China*
[3] *General Administrative Department, Beijing Development Area Co., Ltd., Beijing 100176, China*
[4] *Information and Technology Center, Hebei Normal University for Nationalities, Chengde, Hebei 067000, China*

Correspondence should be addressed to Jingsha He; jhe@bjut.edu.cn

Received 5 February 2013; Revised 21 August 2013; Accepted 21 August 2013

Academic Editor: Muhammad Khurram Khan

In the context of Internet of Things (IoT), multiple cooperative nodes in wireless sensor networks (WSNs) can be used to monitor an event, jointly generate a report and then send it to one or more Internet nodes for further processing. A primary security requirement in such applications is that every event data report be authenticated to intended Internet users and effectively filtered on its way to the Internet users to realize the security of data collection and transmission from the WSN. However, most present schemes developed for WSNs don't consider the Internet scenario while traditional mechanisms developed for the Internet are not suitable due to the resource constraint of sensor nodes. In this paper, we propose a scheme, which we refer to as Data Authentication and En-route Filtering (DAEF), for WSNs in the context of IoT. In DAEF, signature shares are generated and distributed based on verifiable secret sharing cryptography and an efficient ID-based signature algorithm. Our security analysis shows that DAEF can defend against node compromise attacks as well as denial of service (DoS) attacks in the form of report disruption and selective forwarding. We also analyze energy consumption to show the advantages of DAEF over some comparable schemes.

1. Introduction

To become an indispensable part of the Internet of Things (IoT), wireless sensor networks (WSNs) need to adopt IP technologies to create a seamless, global network infrastructure together with the Internet. To achieve this goal, many standardization organizations have been actively pursuing standardization work for creating a global sensor network infrastructure [1, 2]. In the context of the IoT, any IP-enabled node in the Internet shall be able to communicate directly with any remote sensor node in a WSN that is used to monitor specific events. Data transmission from any WSN node to any Internet node can be event-driven, whose scenarios include events that may be sensed sporadically by multiple cooperative sensor nodes when something happens, for example, when detecting fire or door-opening, or can be scheduled at predefined intervals, for example, reporting temperature every two hours. To monitor specific events of interest, more than one sensor node can be used to collect data and transmit it via multihop wireless paths to one or more Internet nodes to improve robustness, especially in an environment in which security threats resulting from internal and external attacks due to node compromises is a serious concern. Under such circumstances, it is still required that data report be sent to one or more of the intended Internet users as accurately as possible.

Since sensor nodes in a WSN may be deployed in an unattended environment, as shown in Figure 1, attackers can relatively easily compromise one or more sensor nodes so that they can be used to inject false event data (e.g., compromised node A can report false data for the event) or disrupt the transmission of legitimate event data (e.g., compromised node B can temper or even discard true data for the event within the multi-hop forwarding process). If undetected, such attacks can cause not only the generation of false alarms but also the depletion of limited energy in the sensor nodes. Moreover, Internet users may not be notified

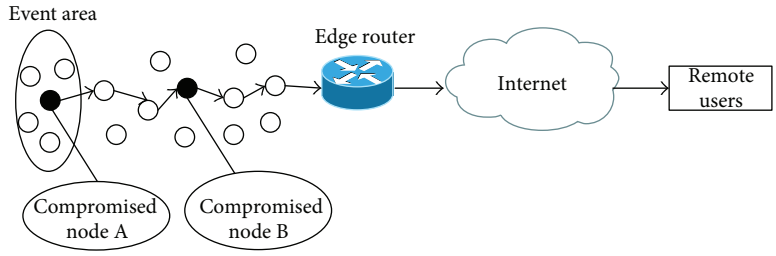


Figure 1: Compromised nodes can inject false event data or disrupt the transmission of legitimate event data.

of a real event quickly and handle the event in time to avoid serious consequences. Therefore, it becomes imperative that security services such as data authenticity and availability be provided to resist such attacks.

In many critical applications, the data that are collected by sensor nodes can also be sensitive. Therefore, it is important to ensure data authenticity to detect the report of false or nonexisting events. If a report for an event can be collectively endorsed by multiple sensor nodes, data authenticity can be ensured in the sense that a certain number of compromised nodes cannot collectively forge a report. That is, to forge a valid report for an event, a larger number of nodes have to be compromised.

Moreover, since denial of service (DoS) attacks can occur as the result of corrupted partial endorsements or discarded true data for the event, authentication alone may not be sufficient. It thus requires that data availability measures be employed in addition to those for authenticity to make the security measures highly resilient to DoS attacks. En-route filtering of false data is vital in saving scarce network resources and in prolonging the life of the network.

Consequently, to achieve security objectives, it is evident that authentication and efficient en-route filtering of data from WSN nodes to one or more Internet users be developed to detect false data injection and to fight against DoS attacks.

Traditional detection mechanisms developed for the Internet usually rely on infrastructure equipment (e.g., firewalls) to filter out distributed denial of service (DDoS) packets, which is deemed to be not adequate for WSNs due to the resource-constrained characteristics of sensor nodes as well as the lack of a comparable infrastructure in WSNs. Meanwhile, current detection mechanisms developed for WSNs rely primarily on the use of predistributed keys shared between sensor nodes and the sink node, which cannot be directly applied to the IoT scenario since the data of the event sensed by cooperative sensor nodes is sent only to one local sink rather than to one or more Internet users who are usually situated in different locations, even in different networks, and who may not be able to establish shared keys with every sensor node to authenticate the event data from a sensor node.

In this paper, we propose a data authentication and en-route filtering (DAEF) scheme to ensure the security of data collection and transmission from WSNs in the context of IoT. In DAEF, we make use of verifiable secret sharing cryptography to distribute the shares based on the most efficient ID-based signature scheme to multiple cooperative sensor nodes. In the case of a node compromise, with the tolerance of an adversary's compromising multiple neighboring nodes in the event area, the event report should be collectively generated, digitally signed, and forwarded to one or more of the intended Internet nodes through multipath routing. The main contributions of this paper are summarized as follows.

(1) After identifying the security requirements on data collection and transmission from WSNs in the context of IoT, we propose a secure and efficient solution to deal with the security problems which have not been sufficiently dealt with in existing solutions.

(2) We propose a data authentication and en-route filtering scheme, referred to as DAEF, without requiring the use of any preshared keys between the Internet users and the sensor nodes. Furthermore, not only can the scheme tolerate node compromise attacks, but it can also mitigate the impact of DoS attacks while minimizing energy consumption for the WSNs.

(3) We analyze some exiting data en-route filtering mechanisms proposed for WSNs with respect to their characteristics and limitations and compare them with DAEF.

(4) We illustrate how DAEF can be used as a secure and efficient mechanism to counter report disruption attacks to WSNs, a capability that we have not found so far in the literature.

(5) We conduct performance comparison between DAEF and some comparable schemes to demonstrate DAEF's advantages over those schemes in terms of energy consumption, making DAEF suitable for WSNs.

The rest of this paper is organized as follows. In the next section, we review some related work on data authentication and en-route filtering. In Section 3, we present our proposed scheme, which includes assumptions, threat model and design goals, two preliminaries, and finally the procedure of the scheme. In Section 4, we analyze our proposed scheme in terms of security and performance and compare it to some comparable schemes. Finally, in Section 5, we conclude this paper in which we also discuss some future work.

2. Related Work

2.1. Authentication Frameworks for IoT. There are currently some authentication frameworks for data reports designed

specifically for WSNs in the IoT scenario. Oliveira et al. proposed the Secure-TWS scheme to authenticate the communication from a single node to multiple users by using certificate-based signature in which the certification authority (CA) is part of the existing infrastructure in the Internet and, hence, is easy to provide since the Internet users only trust the CA and do not have to allow the CA to impersonate as themselves [3]. When the users receive a data report signed by a sensor node, they download the sensor node's public key and the corresponding certificate from the CA to authenticate the report using signature verification. Yasmin et al. proposed a framework for authenticated broadcast/multicast by the sensor node using the identity-based online/offline signature (IBOOS) scheme [4]. The offline phase performs most of the signature computations to calculate the partial signature which is stored on sensor nodes. Whenever a sensor node reports an event, it performs minor computations to obtain the final signature based on the partial signature stored on it.

The above two schemes can enable all sensor nodes in the WSNs to send messages to report critical situations and allow every node on the path from the sender node to the receiver users to verify and filter out false data as early as possible without using any shared keys. The computation overhead of the first scheme is lower than that of the second, but it requires higher communication cost due to the transmission of certificates.

However, these two schemes do not take into consideration of the existence of compromised nodes that may inject false event data as well as can disrupt the transmission of legitimate event data. Firstly, an event may be reported by a single sensor node which may have been compromised but not yet detected, the false report can get propagated to the users who may then mistakenly take incorrect measures. Secondly, should there be a compromised node in the routes to the Internet users, the users might be misled or might even not be able to receive any messages. Therefore, it is necessary to use multiple surrounding sensor nodes to collectively generate a legitimate data report which should also be forwarded to the Internet users via multipath routing.

2.2. Authentication Based on Symmetric Cryptography in WSNs. In WSNs, the problem of authenticating an event report collected by multiple sensors to the local sink node has attracted a great deal of attention in recent years [5–20]. Most of the schemes achieve the goals through using message authentication codes (MAC) based on symmetric keys. The basic idea is to attach MACs to the event reports and to ensure that a legitimate report must have a certain number of valid MACs. When the event report is forwarded to the sink along a routing path, intermediate nodes can detect and drop a forge report if it does not carry enough number of valid MACs. SEF [5] and IHA [6] are two such schemes for filtering injected false data in WSNs. SEF allows both the sink node and the en-route nodes to authenticate a report that has T MACs generated by a cluster attached to it with a certain probability by using the keys from different partitions in a global key pool. IHA verifies a report that has T ($T = t + 1$) MACs and one compressed MAC attached that are computed by the cluster lead node in a deterministic and hop-by-hop fashion through using pairwise keys between two upper or lower associated nodes that are T hops away. All the following schemes are based on this *T-authentication* fashion, but on different technologies to be achieved. In DSF [7], dual key-sharing is used, that is, the random keys sharing and the associated keys sharing, to reduce the number of hops for the forwarded false data. In RAS [8], dynamic authentication tokens from one-way hash chain are used based on a predesigned partition-overlapping key pool scheme. In KAEF [9], a one-way key chain authentication method is used to generate and verify endorsements for transferred data. In STEF [10], the query response operation mode is adopted and the concept of ticket is proposed based on lightweight one-way functions so that messages are only forwarded if they contain a valid ticket that is originally issued by the base station. In PCREF [11], polynomials stored in each node are adopted, including an authentication polynomial and a check polynomial derived from the primitive polynomial, and used for endorsing and verifying the reports. In DEFS [12], the so-called Hill Climbing approach is used to ensure that nodes close to a source cluster hold more keys for the source cluster than those that are further away to balance the network. In GPREF [13], a multiaxis division based approach for deriving location-aware keys is used. In DREF [14], an authentication scheme capable of filtering invalid messages, called CFA [21], and a novel idea of embedding proximity information into a Bloom filter prepared for the query purpose are employed. In EAB [15], an en-route authentication bitmap is developed by using the Bloom filter techniques to build an authentication manifest. In BECAN [16], a bandwidth-efficient cooperative authentication scheme is proposed based on random graph characteristics of sensor node deployment and the cooperative bit-compressed authentication technique. Last, but not the least, in LBRS [17] and LEDS [18], location-based keys are utilized to authenticate a data report to prevent compromised nodes from breaking the entire WSN even though a certain area of the WSN may have been affected.

2.3. Authentication Based on Asymmetric Cryptography in WSNs. Some asymmetric cryptographic schemes, such as CCEF [19], PDF [20], and LTE [22], that rely on signature approaches can enable any report, not just the report that is sent to the sink node, to be authenticated and en-route filtered. Moreover, they do not require any preshared keys. CCEF employs a commutative cipher based on public key cryptography filtering mechanism in which cluster nodes can establish a secret association with the sink on a per-session basis, while the en-route nodes are equipped with a witness key to be used to verify the authenticity of reports without knowing the original session key. PDF leverages Shamir's threshold cryptography and elliptic curve cryptography (ECC) to reject false data packets while LTE makes use of identity-based cryptography (IBC) based on bilinear pairing to bind the private key of each sensor node to both its identity and geographic location.

2.4. Critical Analysis and Comparison of Existing Schemes. Although the existing schemes can effectively perform the

functionality of authentication and false data report filtering, there are still some limitations in different aspects, such as key-sharing limitation, T-threshold limitation, node location limitation, static route limitation, lack of tolerance to report disruption attacks, lack of tolerance to selective forwarding attacks, to name a few. The notion of such limitations is discussed below.

(1) Key-sharing limitation refers to the requirement that every node that generates a report must share key material with the sink.

(2) T-threshold limitation refers to the situation in which the compromise of more than T-authentication partitions would put the whole WSN in danger. Such a limitation makes the network less resilient to the increase in the number of compromised nodes.

(3) Node location limitation refers to the requirement that each node be equipped with GPS capability to locate itself since only rough estimation on the location can be achieved for sensor nodes that are not equipped with GPS.

(4) Static route limitation refers to the reality in which reports get forwarded to a fixed sink along a preestablished path and each en-route node is associated with the source nodes in the event area.

(5) Lack of tolerance to report disruption attacks refers to the situation in which intentional submission of corrupted partial MACs or signatures by compromised sensor nodes would disrupt the process of data filtering by some sensor nodes on the forwarding route, also called packet pollution attacks or false-endorsement based DoS (FEDoS) attacks [23].

(6) Lack of tolerance to selective forwarding attacks refers to the situation in which one or more compromised forwarding sensor nodes can drop a legitimate report, also called path-based DoS (PDoS) attacks [24].

Table 1 provides an analysis on the limitations of some of the main existing schemes. We can see from the table that all symmetric key-based schemes exhibit the key-sharing limitation since it is practically infeasible for every remote user to establish a shared key with each and every one of the sensor nodes. Therefore, these symmetric keys based schemes are not suitable for WSNs in the context of IoT. A result, only PDF and LTE that are asymmetric cryptography-based signature schemes can be used for event data authentication and filtering in the IoT scenario. However, PDF suffers from the vulnerability of both report disruption attacks and selective forwarding attacks. Only DREF, LEDS, and LTE can resist report disruption attacks and selective forwarding attacks that result from node compromises in the event area and from the use of a single route from a sensor node to the sink node. The result of the analysis shows that only LTE is applicable to the IoT scenario and can provide some level of tolerance to both types of DoS attacks. Therefore, we will compare our proposed scheme in terms of performance only to LTE at the same security level.

2.5. Location-Based Threshold-Endorsement (LTE) Mechanism. Due to its functional capability as well as its applicability to the IoT scenario, we would like to elaborate a little more on the LTE scheme. In LTE, the sensor network is divided into $M * N$ square cells of equal side length r. Each cell is labeled with a pair of integers $\langle m, n\rangle$, where $1 \le m \le M$ and $1 \le n \le N$. $\langle X_0, Y_0\rangle$ is the location of the sink. The cell key of cell $\langle m, n\rangle$ is $K_{m,n} = kH(m \parallel n)$ where k is the network master secret key. Let $ID^i_{m,n}$ denote the ith node with location $l^i_{m,n}$ in cell $\langle m, n\rangle$. LTE utilizes the secret sharing technique to assign a share of $K_{m,n}$ to each $ID^i_{m,n}$; that is, $ID^i_{m,n}$ has the share $K^i_{m,n}$.

An event occurs in cell $\langle m, n\rangle$ and is detected by $s \ge t$ nodes. The lead node AP in cell $\langle m, n\rangle$ chooses a random $\alpha \in Z^*_q$ and computes $\theta = e(W, W)^\alpha$ to be broadcast to the other detecting nodes. Upon receiving θ, each detecting node $ID^i_{m,n}$ endorses the report Λ by computing $U^i_{m,n} = K^i_{m,n}h(\Lambda \parallel \theta)$. The node then sends to AP $U^i_{m,n}$ encrypted and authenticated with the pairwise key shared with the AP. Once receiving t or more such endorsements, AP would randomly select t endorsers denoted by a set notation Ω which may include itself. AP would then calculate $U_{m,n} = \sum_{i\in\Omega} \lambda_i U^i_{m,n} = K_{m,n}h(\Lambda \parallel \theta)$ and $\Upsilon_{m,n} = U_{m,n} + \alpha W$. The final report is $\langle \Lambda, \Upsilon_{m,n}, h(\Lambda \parallel \theta)\rangle$. Once deriving $U_{m,n}$, AP verifies its authenticity by checking if equation $e(U_{m,n}, W) = (v^{(0)}_{m,n})^{h(\Lambda\|\theta)}$ holds, where $v^{(0)}_{m,n} = e(K_{m,n}, W)$. If the check does not succeed, AP should proceed to verify each received $U^i_{m,n}$ by checking if (1) holds. Consider the following:

$$e\left(U^i_{m,n}, W\right) = \prod_{j=0}^{t-1}\left(v^{(j)}_{m,n}\right)^{(ID^i_{m,n}\|l^i_{m,n})^j \cdot h(\Lambda\|\theta)}. \quad (1)$$

If the check succeeds, AP considers node $ID^i_{m,n}$ legitimate. AP is therefore able to pinpoint all the endorsers offering false signatures and delete them from Ω.

AP then sends to the sink the final report along a multihop path discovered using a secure multipath routing protocol. Upon receiving a report $\langle \Lambda, \Upsilon_{m,n}, h(\Lambda \parallel \theta)\rangle$ to be forwarded, each intermediate node computes

$$\theta' = e\left(\Upsilon_{m,n}, W\right) e\left(H\left(m \parallel n\right), -W_{\text{pub}}\right)^{h(\Lambda\|\theta)}, \quad (2)$$

where $W_{\text{pub}} = kW$ is a public system parameter. If the report is authentic, then $\theta' = \theta$. Therefore, if $h(\Lambda \parallel \theta') = h(\Lambda \parallel \theta)$, then an intermediate node would consider the report authentic and hence forward it to the next hop.

However, LTE is not always feasible because it requires that every sensor node be equipped with localization capability, which incurs extra communication overhead as well as latency. Moreover, the bilinear pairing utilized in LTE is too expensive for low energy sensor nodes. In LTE, a data report must be cosigned by t out of T nodes ($t \le T$) in the event area. Thus, an adversary needs to compromise at least t nodes in order to inject false data. It is worth mentioning, however, that the relationship between t and T is not fully discussed in LTE with respect to dealing with report disruption attacks. From our analysis, we have found that the critical relationship $T - (t - 1) \ge t$, that is, the maximum

Table 1: Analysis of limitations.

Schemes	Key-sharing limitation	T-threshold limitation	Node location limitation	Static route limitation	Lack of tolerance to FEDoS attacks	Lack of tolerance to PDoS attacks
SEF	Yes	Yes	No	No	Yes	Yes
IHA	Yes	Yes	No	Yes	Yes	Yes
DSF	Yes	No	No	No	Yes	Yes
RAS	Yes	No	No	No	Yes	Yes
KAEF	Yes	No	No	No	Yes	Yes
STEF	Yes	No	Yes	Yes	Yes	Yes
PCREEF	Yes	No	No	No	Yes	Yes
DEFS	Yes	Yes	No	No	Yes	Yes
GPREF	Yes	No	Yes	No	Yes	Yes
DREF	Yes	No	No	No	No	No
EAB	Yes	No	No	Yes	Yes	Yes
BECAN	Yes	No	No	No	Yes	No
LBRS	Yes	No	Yes	Yes	Yes	Yes
LEDS	Yes	No	Yes	Yes	No	No
CCEF	Yes	No	Yes	Yes	Yes	Yes
PDF	No	Yes	No	No	Yes	Yes
LTE	No	No	Yes	No	No	No

number of compromised nodes, that is, $t-1$, for the scheme to tolerate false data injection attacks cannot cooperatively cause the report disruption attacks. Thus, we have $T \geq 2t - 1$.

3. The Proposed Scheme

In this section, we describe the proposed DAEF, a data authentication and en-route filtering scheme, that can be used to ensure the security of data collection and transmission from WSNs in the context of IoT. In applications to which DAEF can be applied, we assume that a group of sensor nodes are used to monitor an event. In the proposed DAEF, a group ID-based signature is introduced for each event report so that any intermediate node and any Internet user with the identity of the group lead node can easily verify the event report, which easily removes the key-sharing and the T-threshold limitations, while realizing T-authentication to tolerate node compromise attacks without requiring static route and node localization technology. DAEF also employs the verifiable secret sharing algorithm to distribute and verify signature shares in order to defend against report disruption attacks. Last, but not the least, we employ the most efficient ID-based signature algorithm in DAEF to reduce computation overhead that results from signature generation and verification operations. Moreover, a multi-path routing protocol should be used in DAEF to resist selective forwarding attacks.

3.1. Assumptions. We assume that all the sensor nodes in WSNs are deployed uniformly and bootstrapped securely using the scheme proposed in [25] so that the one-hop neighboring sensor nodes can establish pairwise keys and trust relationships to form a network with multi-hop cluster-tree hierarchical topology. Each node establishes and stores a neighbor trust list as shown in Figure 2. We also assume that every event of interest can be detected by multiple, say T ($T > 1$), sensor nodes in one group, each group covering a detecting area. Then, the event needs to be reported during which a group of at least t ($T \geq 2t - 1$) nearby legitimate nodes should collaboratively agree on the event that will be forwarded to one or more Internet users. In this scenario, T and t are predefined system parameters and are determined by the application requirements. For each group, a lead head is elected and is responsible for collecting and summarizing all the received detection results from detecting nodes in the group, and generating a final report on behalf of the group. This group of neighboring nodes generates and broadcasts the signed report to a lead node which then aggregates the signatures before sending them to one or more Internet users through one or more forwarding nodes.

Moreover, we assume that the Internet users are determined by the service provider (SP) based on the service provided by the sensor nodes. Take the smart home applications as an example, the detection of fire in a home should be reported to the family members, security guards of the community, and the fire department, while the temperature in the home may be reported to all family members every two hours. The corresponding software code can be preloaded in the sensor nodes prior to deployment. Furthermore, the code can be dynamically updated by using an end-to-end secure communication protocol (referring to [26]).

3.2. Threat Model and Design Goals. An adversary can eavesdrop on all traffic, inject packets, replay older packets, and take full control of the compromised nodes to launch false report injection attacks and DoS attacks. In our model, we assume that most T-t neighboring nodes in an event area

2 bytes	2 bytes	16 bytes	2 bits	1 byte	1 byte
Neighbor ID	*Short address*	*Shared key*	*Relationship*	*Hop count*	*Trust value*

Figure 2: Neighbor trust list: *neighbor ID* is the *ID* of the node's neighbor; *short address* is the short address of the node's neighbor; *shared key* is the pairwise key shared between the node and its neighbor; *relationship* points that the neighbor is the node's father, child, sibling, and so on; *hop count* is the distance between the node and the edge route, *trust value* is the computed result making use of the utility value method in a multiple criteria decision making scheme.

can be compromised. Our objective is to design a scheme to detect these attacks for the event report in the IoT scenario. DAEF should achieve the following goals.

(1) It should not require the establishment of preshared keys between Internet users and sensor nodes (to overcome the key-sharing limitation).

(2) It should not depend on the localization technology to prevent compromised nodes from breaking the whole WSN (to overcome the T-threshold limitation).

(3) It should tolerate node compromises in the WSNs even if the locations of the sensor nodes may not be known (to overcome the node location limitation).

(4) It can mitigate the impact of DoS attacks including report disruption attacks and selective forwarding attacks (to overcome the FEDoS and PDoS attacks).

(5) It should keep the overhead of communication and computation as low as possible in the WSNs.

3.3. Preliminaries

3.3.1. ID-Based Signature. As discussed above, ECC-based signature (i.e., ECDSA) requires one point multiplication operation to generate a signature and two point multiplication operations to verify a signature. Moreover, authentication of the public key of the signer also requires two point multiplication operations. For an ECC of 160 bites, ECDSA produces a 60-byte signature, resulting in more than a 160-byte message payload (including a 60-byte ECDSA signature and a certificate with at least 100 bytes). With the current state of the art technology, the most efficient ID-based signature (i.e., vBNN [27]) needs one point multiplication operation to sign a message and three point multiplication operations to verify the signature with the length of a signature being 83 bytes. Let us briefly describe the vBNN scheme below.

Given a sensor node ID_i, the SP picks a random number $r_i \in Z_q$, where the multiplicative group $Z_q = [1, \ldots, q-1]$ computes $R_i = r_i P$, where P is a large prime, and an elliptic curve $E(F_p)$ is defined over a finite field $F_p = [1, 2, \ldots, P-1]$. Then, SP calculates $S_i = r_i + xH_1(ID_i \parallel R_i)$ in which x is the master key of the WSN picked by SP and $H_1 : \{0,1\}^* \times G_1 \rightarrow Z_q$, where G_1 is an additive group of the prime order q. R_i and S_i are stored in the sensor node ID_i.

Given a message M, the signer ID_i performs the following steps to sign the message.

(1) Choose a random number $y \in Z_q$, and compute $Y = yP$.

(2) Compute $h = H_2(ID_i \parallel M \parallel R_i \parallel Y)$ and $z = y + S_i h$, where $H_2 : \{0,1\}^* \rightarrow Z_q$.

(3) (R_i, h, z) is the digital signature.

The signer then sends (M, R_i, h, z) to the receiver. To verify the message and the signature, the receiver does the following steps.

(1) Compute $c = H_1(ID_i \parallel R_i)$.

(2) Check the equation $h = H_2(ID_i \parallel R_i \parallel M \parallel zP - h(R_i + cP_{\text{pub}}))$, where $P_{\text{pub}} = xP$ is a public parameter.

3.3.2. Verifiable Secret Sharing. Sensor node ID_i generates a secret polynomial $f_{S_i}(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1}x^{t-1}$, where $a_0, \ldots, a_{t-1}$ are random numbers picked by the sensor node ID_i and the secret key S_i can be picked as $S_i = a_0$. The secret share of S_i for the neighboring node ID_j is thus $C^j_{S_i} = f_{S_i}(ID_j)$. Then, any t sensor nodes together can reconstruct S_i by using Lagrange interpolation $S_i = \sum_{j=1}^{t} l_j C^j_{S_i}$, where $l_j = \prod_{k=1, k\neq j}^{t}(ID_k/(ID_k - ID_j))$ is the Lagrange coefficient. However, it is computationally infeasible if fewer than t sensor nodes try to reconstruct the secret key S_i. All $C^j_{S_i}$ must be distributed through the secure communication channels. The sensor node ID_i broadcasts $\varepsilon_0 = g^{a_0}$ and $\varepsilon_n = g^{a_n} (\text{mod } p)$ $(n = 1, 2, \ldots, t-1)$, and every ID_j can verify the received $C^j_{S_i}$ by using the equation $g^{C^j_{S_i}} = \prod_{k=0}^{t-1} \varepsilon_k^{ID_j{}^k} (\text{mod } p)$, while ID_i can verify the reconstructed S_i by using the equation $g^{S_i} = \varepsilon_0$.

3.4. The Data Authentication and Filtering Scheme

3.4.1. Initialization. During the bootstrapping phase, the lead node in every group distributes the secret share of S_i to all group nodes. Specifically, in ith group, the lead node ID_i generates a secret polynomial $f_{S_i}(x)$ and distributes the secret share $f_{S_i}(ID_j)$ to every group node ID_j using the shared key between them. Then, ID_i deletes S_i and $f_{S_i}(x)$ but stores $f_{S_i}(ID_i)$. Therefore, ID_i only needs to be authenticated by any other $t-1$ group nodes and get $t-1$ secret shares in order to reconstruct S_i.

3.4.2. Report Generation. When an event occurs, the lead node will prepare a report, say, E. To get an agreement on the event from other group nodes, the lead node ID_i broadcasts E to all the group nodes and authenticates itself to them. After receiving E, a group node will find the difference between the received E and what it has sensed. If the difference is

within a predefined error range, it will agree on E and endorse the signature. These T group members including ID_i itself, taking one node ID_j as an example, will generate one random polynomial $f_{y_i}^j(x)$ as shown in (3), in which the share of the random number is $C_{y_i}^j(k) = f_{y_i}^j(ID_k)$ $(k = 1, 2, \ldots, T)$, distribute $C_{y_i}^j(k)$ to the other sensor nodes ID_k, and broadcast $\varepsilon_{j_n} = g^{a_{j_n}} (\mathrm{mod} p)$ $(n = 0, 1, 2, \ldots, t-1)$. Consider the following:

$$f_{y_i}^j(x) = a_{j_0} + a_{j_1} x + a_{j_2} x^2 + \cdots + a_{j_{t-1}} x^{t-1}. \quad (3)$$

Note that the distribution of secret shares should be protected using the shared keys between the communicating peers. Therefore, each node will receive no less than $m-1$ $(t \le m \le T)$ secret shares which should be verified as discussed in Section 3.3.2. For example, ID_k should verify $C_{y_i}^j(k)$ received from ID_j according to formula (4). Consider the following:

$$g^{C_{y_i}^j(k)} = \prod_{n=0}^{t-1} \varepsilon_{j_n}^{ID_k^n} \left(\mathrm{mod} p\right). \quad (4)$$

During the verification phase, a compromised group member may be detected. If ID_j find a corrupted partial secret share sent by a group member, ID_j broadcasts the ID of the compromised group member. If more than t members claim that one node has been compromised, each legitimate sensor node can find all compromised group members and no less than t legitimate group members. These m legitimate sensor nodes should use the verified $m-1$ secret shares and its own share to compute the share of y_i, denoted as y_i^j, shown in (5). Consider the following:

$$\begin{aligned} y_i^j &= \sum_{n=1}^{m} f_{y_i}^n \left(ID_j\right) \\ &= \left(\sum_{n=1}^{m} a_{n0}\right) + \left(\sum_{n=1}^{m} a_{n1}\right) ID_j + \left(\sum_{n=1}^{m} a_{n2}\right) ID_j^2 \\ &\quad + \cdots + \left(\sum_{n=1}^{m} a_{nt-1}\right) ID_j^{t-1}. \end{aligned} \quad (5)$$

The secret random number y_i is generated from the polynomial $f_{y_i}(x) = \sum_{n=1}^{m} f_{y_i}^n(x)$, shown in formula (6), which is endorsed by these m legitimate sensor nodes and $y_i = f_{y_i}(0)$. Therefore, no sensor node knows y_i, and any t sensor nodes can reconstruct y_i using Lagrange interpolation: $y_i = \sum_{j=1}^{t} l_j y_i^j$, in which $l_j = \prod_{n=1, n \ne j}^{m} (ID_n / (ID_n - ID_j))$. Consider

$$\begin{aligned} f_{y_i}(x) &= \sum_{n=1}^{m} f_{y_i}^n(x) \\ &= \left(\sum_{n=1}^{m} a_{n0}\right) + \left(\sum_{n=1}^{m} a_{n1}\right) x + \left(\sum_{n=1}^{m} a_{n2}\right) x^2 \\ &\quad + \cdots + \left(\sum_{n=1}^{m} a_{nt-1}\right) x^{t-1}. \end{aligned} \quad (6)$$

Each legitimate neighboring node ID_j except the lead node ID_i sends $l_j y_i^j P$ to ID_i which then sums up the received m shares to get $Y_i = \sum_{j=1}^{m} l_j y_i^j P$ and broadcasts Y_i along with R_i. Each legitimate neighboring node ID_j including the lead node ID_i computes $z_j = l_j y_i^j + l_j C_{S_i}^j h$, where $h = H_2(ID_i \parallel E \parallel R_i \parallel Y_i)$ and sends z_j to ID_i which then sums up z (7). It is possible that some of the neighboring nodes have been compromised during this phase and, thus, may provide the lead node ID_i with incorrect signatures. Therefore, ID_i should verify their authenticity by checking the equation $Y_i = zP - h(R_i + cP_{\mathrm{pub}})$ where $c = H_1(ID_i \parallel R_i)$. Consider the following:

$$z = \sum_{j=1}^{m} \left(l_j y_i^j + l_j C_{S_i}^j h\right) = \sum_{j=1}^{m} l_j y_i^j + \sum_{j=1}^{m} l_j C_{S_i}^j h = y_i + S_i h. \quad (7)$$

Finally, ID_i broadcasts the final data report (E, R_i, h, z) and assigns multiple upstream nodes in its neighbor trust list to make the report forwarded to the Internet users through multipath routing. In the cases in which the compromised lead node may either not send the final report or transmit a bogus report with a wrong (E, R_i, h, z), it will be detected by all legitimate group nodes. The verification is the same as the en-route filtering operations to be described in Section 3.4.3. In this case, the legitimate neighboring nodes will randomly elect a new lead node among themselves to generate a new threshold-endorsement and send the final report to the Internet users. The whole report generation procedure is illustrated in Algorithm 1.

3.4.3. En-Route Filtering of Data Report. We denote P_f as the en-route verification probability. The forwarding sensor node verifies the signature of a report with the probability P_f which is a predefined system parameter. As discussed in Section 3.3.1, the verifying intermediate node or the final Internet user first computes $c = H_1(ID_i \parallel R_i)$, then checks (8). The data report will be regarded as authentic and forwarded to multiple upstream nodes in their neighbor trust lists if the verification is successful, otherwise, it will be immediately discarded. Consider

$$h = H_2\left(ID_i \parallel R_i \parallel E \parallel zP - h\left(R_i + cP_{\mathrm{pub}}\right)\right). \quad (8)$$

4. Analysis

4.1. Security Analysis

4.1.1. Resilience to Node Compromise Attacks. DAEF uses the threshold signature generation to sign any event report. The attacker must know the private key S_i and the secret random number y_i to forge the signature of ID_i by compromising at least t nodes in the event area. In some cases, the lead node itself may have been compromised, resulting in a higher level of risk. However, DAEF does not reveal any $C_{S_i}^j$ and y_i^j at any step. Firstly, the initialization phase is secure in which the lead node distributes the shares of S_i within a short time after bootstrapping. Secondly, each neighboring node submits

for (each sensor ID_j, $j = 1, 2, \ldots, T$)
ID_j: $C^j_{y_i}(k) = f^j_{y_i}(ID_k)$ $(k = 1, 2, \ldots, T)$
$ID_j \rightarrow ID_k$: $C^j_{y_i}(k)$
$ID_j \rightarrow {}^*$: $\varepsilon_{j_n} = g^{a_{j_n}} \pmod p$ $(n = 0, 1, 2, \ldots, t-1)$
for (each sensor ID_j, $j = 1, 2, \ldots, T$)
ID_j: $g^{C^j_{y_i}(k)} = \prod_{n=0}^{t-1} \varepsilon_{j_n}{}^{ID_k{}^n} \pmod p$
ID_j: $y^j_i = \sum_{n=1}^{m} f^n_{y_i}(ID_j)$
$ID_j \rightarrow ID_i$: $l_j y^j_i P$
$ID_i \rightarrow {}^*$: $Y_i = \sum_{j=1}^{m} l_j y^j_i P, R_i$
for (each sensor ID_j, $j = 1, 2, \ldots, m$)
$ID_j \rightarrow ID_i$: $z_j = l_j y^j_i + l_j C^j_{S_i} h$
ID_i: $z = \sum_{j=1}^{m} \left(l_j y^j_i + l_j C^j_{S_i} h\right)$
$ID_i \rightarrow {}^*$: (E, R_i, h, z)

Algorithm 1: Report generation procedure.

$l_j y^j_i P$ rather than y^j_i, making it impossible for the lead node to derive y^j_i from $y^j_i P$ due to the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP). Thirdly, $z_j = l_j y^j_i + l_j C^j_{S_i} h$ that is submitted by each neighboring node has two unknown numbers y^j_i and $C^j_{S_i}$, so the lead node cannot derive y^j_i and $C^j_{S_i}$ from z_j. Finally, should the lead node change the report with a wrong (E, R_i, h, z) in the final step, it would be detected by the forwarding nodes as well as by all the legitimate neighboring nodes. These nodes can then elect a new lead node to generate a new threshold-endorsement and send the final report to the Internet nodes. In the worst case, even if the attacker can derive the private key S_i by compromising t nodes in the event area, it will not affect any other groups.

4.1.2. Mitigation of Report Disruption Attacks. DAEF leverages verifiable Shamir's secret sharing cryptography described in Section 3.3.2 which has been shown to be secure [28]. In the report generation phase, at least t legitimate sensor nodes cooperatively generate the secret random number y_i by exchanging the shares $C^j_{y_i}(k) = f^j_{y_i}(ID_k)$, which may be disrupted by compromised nodes. For example, an attacker may provide an incorrect share $C^j_{y_i}(k)$ to a neighboring node so that it will not get the right share $y^j_i = \sum_{k=1}^{m} f^k_{y_i}(ID_j)$. However, in DAEF, each node will verify the received shares, detect compromised neighboring nodes, and broadcast the detection result so that only legitimate shares will be used in the y^j_i and only the shares z_j computed by legitimate neighboring nodes can be used by the lead node. Note, however, that if the attacker only distributes the wrong shares to only some of the legitimate neighboring nodes that are able to detect the wrong shares, based on the detected results broadcast by such neighboring nodes, it is not possible for the legitimate neighboring nodes to identify the compromised nodes.

4.1.3. Mitigation of Selective Forwarding Attacks. In DAEF, in order to mitigate selective forwarding attacks, the lead node and the intermediate nodes would forward the final report (E, R_i, h, z) to multiple upstream nodes that are in their neighbor trust lists to ensure that the report is forwarded to the Internet users through multipath routing. Unless all the forwarding nodes are compromised, the legitimate report will ultimately be delivered to the destinations. However, this solution will incur high communication overhead. In the worst case in which all the upstream nodes of a forwarding node are compromised, another route path should be used by using a secure multi-path routing protocol in the WSN, such as SPREAD [29], which is out of the scope of this paper.

4.2. Performance Analysis. In this section, we evaluate the performance of DAEF in terms of filtering efficiency, the number of hops that false data can travel, and the ratio of compromised area, which are the main metrics for the evaluation of en-routing filtering schemes [30]. We also analyze and compare DAEF to LTE in terms of computation, communication, and energy consumption. We conducted all experiments in the analysis, evaluation, and comparison by using MATLAB [31] plus some programming in VC^{++} whenever necessary.

4.2.1. General Analysis. First, let us analyze filtering efficiency P_h which is defined as the probability of successful filtering of false data within a specified number of hops. Similar to LTE, the probability of false data that can be filtered out within h hops is

$$P_h = 1 - \left(1 - P_f\right)^h. \tag{9}$$

Clearly, the greater the value of P_f, the greater the P_h can achieve, that is, the better the filtering efficiency. However, a smaller P_f can lower the computation overhead of intermediate nodes. Figure 3 shows the filtering efficiency with different P_f from which we can see that when P_f is 0.5, more than 90% of false reports can be filtering out within 4 hops. Ever for a small P_f, say 0.2, less than 11% of false reports can travel over 10 hops. Therefore, for a large WSN which has long

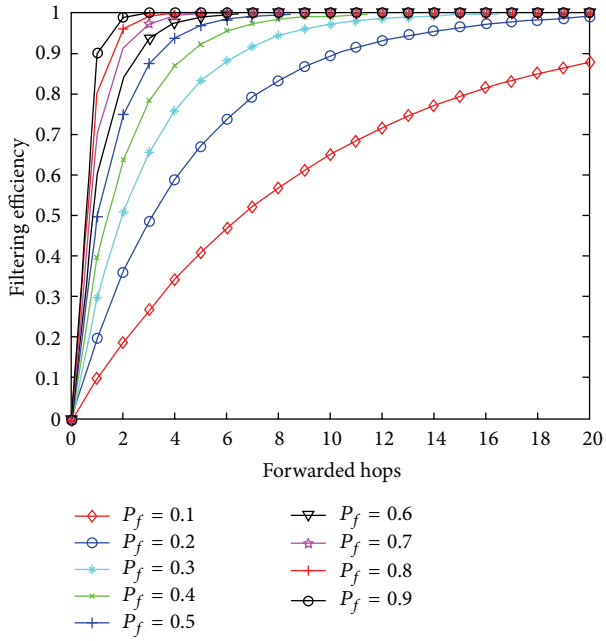


Figure 3: The filtering efficiency with different P_f.

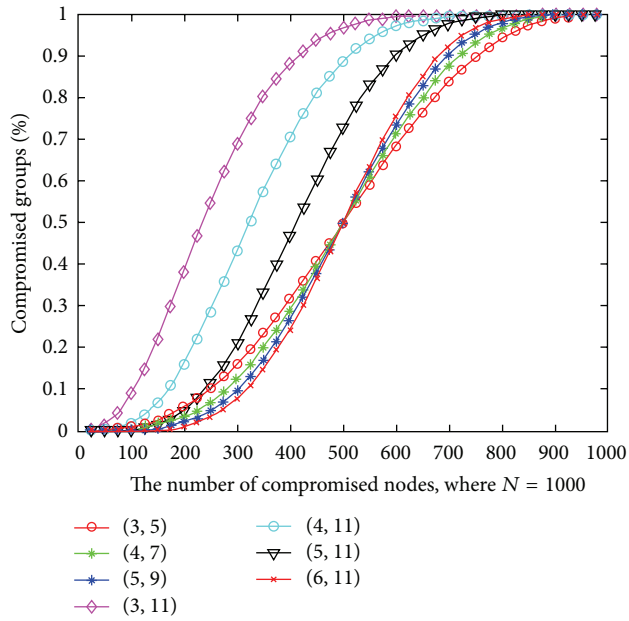


Figure 4: The percentage of compromised groups with the increased number of compromised nodes when $N = 1000$.

forwarding paths, DAEF can efficiently filter out false reports as early as possible to save the energy of legitimate nodes.

The number of hops that false data can travel $E(h)$ is defined as the average hops that false data are forwarded before being filtered and dropped, which reflects the filtering effectiveness. Similar to LTE, the average number of hops that a false report is forwarded before being filtered out is

$$E(h) = \sum_{i=1}^{\infty} iP_f\left(1 - P_f\right)^{i-1} = \frac{1}{P_f}. \tag{10}$$

The ratio of compromised area Ps is defined as the percentage of compromised sensor nodes in the terrain, which reflects the effectiveness of filtering resilience to the increase on the number of compromised nodes. Given that the network size is N and that the average number of nodes in each group is T, when the attacker successfully compromises Nc sensor nodes, the probability that no node in an event group is compromised is

$$P_{(0)} = \frac{C_{N-T}^{Nc}}{C_N^{Nc}}. \tag{11}$$

Let $P_{(i)}$ represent the probability that i nodes are compromised in a group. Then,

$$P_{(i)} = \frac{C_T^i C_{N-T}^{Nc-i}}{C_N^{Nc}}. \tag{12}$$

Therefore, the percentage of compromised nodes is

$$Ps = 1 - \sum_{i=0}^{t-1} P_{(i)} = 1 - P_{(0)} - \sum_{i=1}^{t-1} \frac{C_T^i C_{N-T}^{Nc-i}}{C_N^{Nc}}. \tag{13}$$

Figure 4 shows the resilience as the number of compromised nodes increases. We can observe that the percentage of compromised groups increases very slowly with an increasing number of compromised nodes when less than 20% of all the nodes are compromised. Even if the number of compromised nodes reaches 200, only 10% of groups are compromised in all the cases except $(t, T) = (3, 11)$ and $(t, T) = (4, 11)$.

4.2.2. Comparison Analysis. We analyze DAEF and LTE in terms of computation cost and communication overhead and then compare them in terms of energy consumption to show the advantages that DAEF has over LTE.

First, we perform the analysis on computation cost. In all the schemes, expensive operations are normally pairing (Pa), point multiplication (Pm), and exponentiation (Exp). In DAEF, each endorsing sensor node needs to do one point multiplication operation to generate $l_j y_i^j P$ and $T \times t$ exponentiation operations which include t exponentiation operations to generate ε_n $(n = 0, 1, 2, \ldots, t-1)$ and $(T-1) \times t$ exponentiation operations to verify $T - 1$ received shares. Thus, the total number of computational operations in an event group is $T * Pm + T^2 t *$ Exp. Meanwhile, each verifying forwarding sensor node needs three point multiplication operations to authenticate an event report. In contrast, in LTE, the number of computational operations in an event group is $2 * Pa + (T + 1) * Pm + 2 *$ Exp in cases in which no compromised nodes are selected by the lead node but is $(T + 1) * Pa + (T + 1) * Pm + (Tt - t + 2) *$ Exp in the worst case. In addition, each forwarding sensor node needs two pairing operations and one exponentiation operation to authenticate an event report.

For communication overhead, which is defined as the total number of messages generated in an event group, in DAEF, the T sensor nodes have to jointly generate a random number y_i for each report, which contributes most to the

communication overhead because the private key S_i can be distributed during the bootstrapping phase. For a share y_i, each sensor node needs to send $T-1$ secret shares to the $T-1$ neighboring nodes and broadcast one promise ε_n ($n = 0, 1, 2, \ldots, t-1$) and one detection result. In the signature generation phase, each legitimate group sensor node sends $l_j y_i^j P$ and z_j to the lead node and the lead node broadcasts Y_i along with R_i. Thus, the total number of messages required to generate a signature for an event report is $(T+t+1)T$. In contrast, in LTE, each endorsing node only needs to send one share $U_{m,n}^i$ to the lead node which should broadcast θ. Thus, the total number of messages required to generate a signature for an event report is T.

Note that in the above analysis, we assume that there is no compromised node in any of the event groups. Let us now analyze and compare DAEF to LTE in terms of energy consumption for the whole WSN under the assumption that there is at least one compromised node in an event group at the time of reporting an event, for these two schemes offer a similar level of security and both can deal with report disruption attacks and selective forwarding attacks.

We employ the similar model to LTE when performing analysis and comparison on energy consumption which is determined by communication overhead as well as computation cost. We assume that the sensor nodes have the same capabilities as those of a standard Crossbow's MICA2 mote [32] which has 8-bit ATmegal 128L clocked at about 7.37 MHz microcontroller and complies with the IEEE 802.15.4 standards with data transmission rate of 12.4 kbps. According to [4], completing a 160-bit point multiplication operation of ECC, a pairing operation, and an exponentiation operation consumes 24.3 mJ, 62.73 mJ and 2.81 mJ, respectively. In addition, MICA2 consumes 52.2 μJ, and 19.4 μJ to transmit and to receive one byte, respectively. We assume that the length of the node's *ID* is 2 bytes, making the lengths of $C_{y_i}^j(k)$, ε_n, $l_j y_i^j P$, z_j, Y_i and R_i to be $2t$ bytes, $20t$ bytes, 40 bytes, 20 bytes, 40 bytes, and 40 bytes, respectively. The original report is assumed to be 15 bytes, thus allowing us to transmit a report in one data packet. We denote ε as the average number of hops a report travels in the WSNs. For the sake of simplicity, we only consider the single routing path; thus, both schemes involve εP_f en-route filtering operations.

In the analysis, we specify (t, T) to be $(2, 3)$, $(3, 5)$, $(4, 7)$, $(5, 9)$, and $(6, 11)$, respectively, because of the relationship between t and T discussed in Section 2.4. Figure 5 shows the energy consumption for various ε when $P_f = 0.2$ under the condition that the number of compromised nodes in the event group is the maximum value $t-1$. We can see from the figure that energy consumption in DAEF is lower than that in LTE for $(2, 3)$, $(3, 5)$, $(4, 7)$, and $(5, 9)$. However, the difference narrows as the value of T increases until when (t, T) is larger than $(6, 11)$ where DAEF will consume more energy than LTE. In addition, when ε increases by one hop, the increase in energy consumption in DAEF is about 15 mJ whereas that in LTE is more than 25 mJ. That is because the energy consumption of computation in DAEF that incurs $O(t)$ cost of point multiplication and $O(t^2T)$ cost of exponentiation for report generation and three point multiplication operations for report en-route verification is lower than that in LTE that incurs $O(T)$ cost of pairing and $O(t)$ cost of point multiplication as well as $O(tT)$ cost of exponentiation for report generation and two pairing operations and one exponentiation operation for report en-route verification. However, the energy consumption of communication in DAEF for generating $O(t^2, T)$ communication cost is higher than that in LTE which only generates $O(t, T)$ communication cost.

Figure 6 shows the energy consumption for various P_f when $\varepsilon = 10$ under the condition that the number of compromised nodes in the event group is the maximum value $t-1$. With the same (t, T), the difference in energy consumption gets higher as P_f increases. This is because one filtering operation requires two pairings and one exponentiation in LTE which incurs more cost of computation, about 128 mJ, than three point multiplications, about 72 mJ, of one filtering operation in DAEF.

Figure 7 shows the energy consumption as the number of compromised nodes increases in an event group when $P_f = 0.2$ and $\varepsilon = 10$. We can see from the figure that both DAEF and LTE perform better as the number of compromised nodes increases with the same (t, T). This is because the number of legitimate nodes will decrease as the number of compromised nodes increases, hence reducing the computation cost and communication overhead in both schemes. Meanwhile, in DAEF, the larger the value of T is, the larger the difference is for the same (t, T). In LTE, however, the reduction in energy consumption becomes less significant as the value of T increases. That is because the change in the communication overhead in DAEF is more than that in LTE in terms of the number of compromised nodes, which can be seen when $(t, T) = (5, 9)$.

Figure 8 shows the energy consumption for various parameter t when $P_f = 0.2$ and $\varepsilon = 10$ under the condition that the number of compromised nodes in the event group is maximum value $t-1$. The energy consumption in DAEF is lower than that in LTE except for the values $T \geq 11$ and $t > 4$. However, the former increases slowly while the latter decreases as t increases. In LTE, regardless of the value of t, the computation cost has little difference, which results from the number of valid shares $U_{m,n}^i$ randomly selected by the lead node. Moreover, the larger value of t, the more effective computation and communication get, which reduces the waste of energy. In DAEF, the slowly increasing energy consumption results from the computation and broadcasting of promise ε_n ($n = 0, 1, 2, \ldots, t-1$).

In conclusion, DAEF outperforms LTE for data authentication and en-route filtering when an event group has a smaller number of nodes and larger number of compromised nodes.

5. Conclusion

In this paper, we proposed DAEF, a new data authentication and filtering scheme, to ensure the security of data collection and transmission from WSNs in the context of IoT. In the scheme, the verifiable secret sharing cryptography is used for the distribution of the shares to multiple

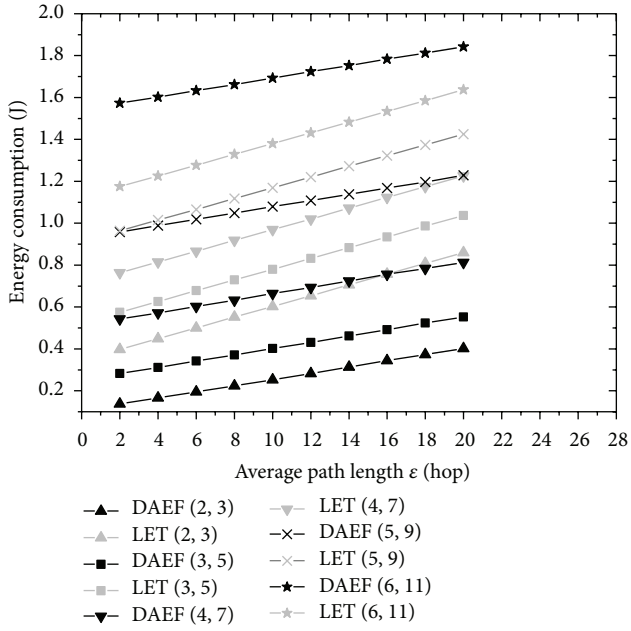


Figure 5: Energy consumption for various ε when $P_f = 0.2$.

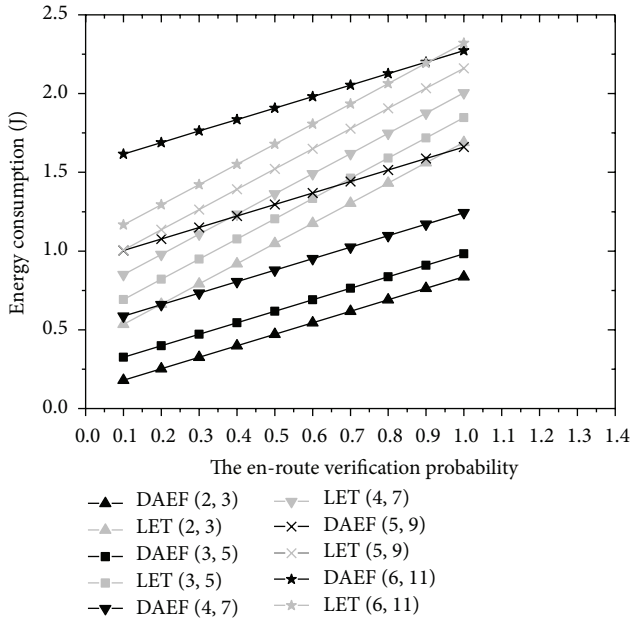


Figure 6: Energy consumption for various P_f when $\varepsilon = 10$.

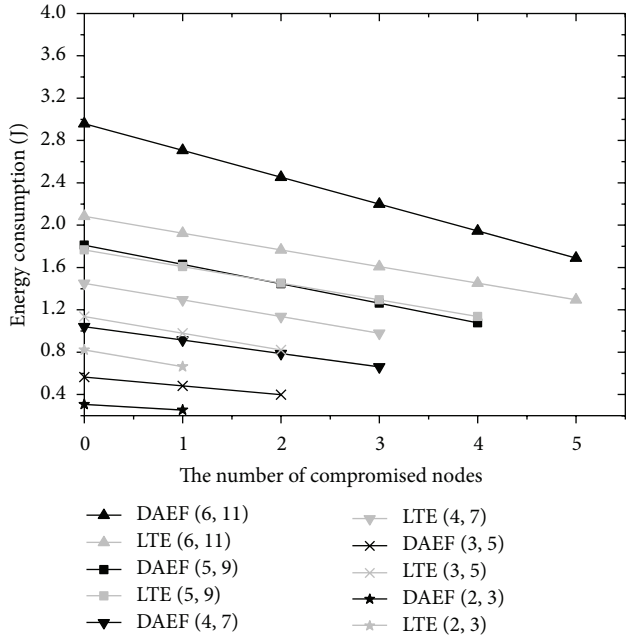


Figure 7: Energy consumption with an increasing number of compromised nodes for various (t, T) when $P_f = 0.2$ and $\varepsilon = 10$.

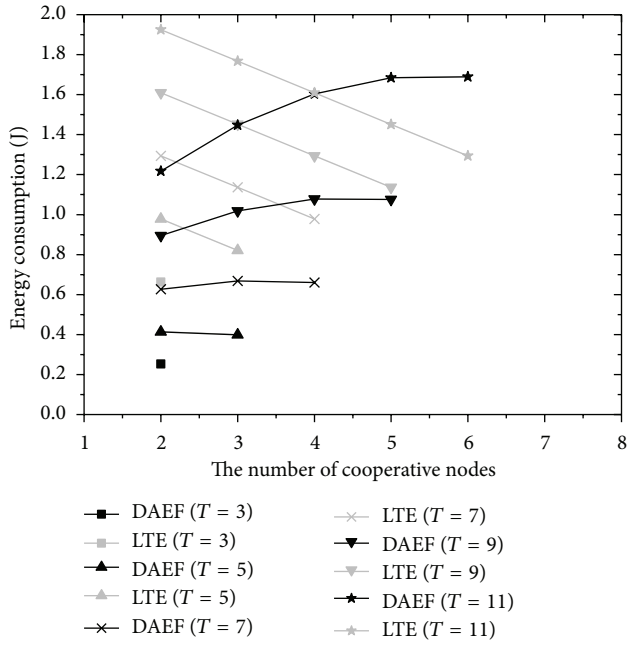


Figure 8: Energy consumption for various t when $P_f = 0.2$ and $\varepsilon = 10$.

neighboring collective sensor nodes based on the most efficient *ID*-based signature scheme. As long as an adversary does not compromise more than T-t neighboring nodes in an event area, any event report can be collectively generated with a digital signature attached and forwarded via multipath routing to multiple Internet nodes. Analysis on the proposed scheme showed that DAEF can effectively defend against node compromised attacks and DoS attacks in the forms of report disruption attacks and selective forwarding attacks. Quantitative analysis to compare DAEF to the LTE scheme has also been performed in terms of energy consumption in which we showed that DAEF outperforms LTE in terms of energy consumption when fewer numbers of nodes and more numbers of compromised nodes of a group exist in the event group. In the future, we will conduct more experiment in real network settings to verify the results and to further improve the performance in terms of latency in DAEF.

Acknowledgments

The work in this paper has been supported by funding from National Natural Science Foundation of China (61272500) and from Beijing Education Commission Science and Technology Fund (KM201010005027).

References

[1] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals," IETF RFC 4919, Internet Engineering Task Force, 2007.

[2] G. Montenegro, N. Kushalnagar, and J. Hui, "Transmission of IPv6 packets over IEEE 802.15.4 networks," IETF RFC 4944, Internet Engineering Task Force, 2007.

[3] L. B. Oliveira, A. Kansal, C. P. L. Gouvêa et al., "Secure-TWS: authenticating node to multi-user communication in shared sensor networks," *The Computer Journal*, vol. 55, no. 4, pp. 384–396, 2012.

[4] R. Yasmin, E. Ritter, and G. Wang, "An authentication framework for wireless sensor networks using identity-based signatures," in *Proceedings of the 10th IEEE International Conference on Computer and Information Technology (CIT '10)*, pp. 882–889, Bradford, West Yorkshire, UK, July 2010.

[5] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 2446–2457, Hong Kong, China, March 2004.

[6] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 259–271, Oakland, Calif, USA, May 2004.

[7] Q. Sun and M. Wu, "A double key-sharing based false data filtering scheme in wireless sensor networks," in *Proceedings of the 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom '11), 8th IEEE International Conference on Embedded Software and Systems (ICESS '11), 6th International Conference on Frontier of Computer Science and Technology (FCST '11)*, pp. 509–516, Changsha, China, November 2011.

[8] Y. Hu, Y. Lin, Y. Liu, and W. Zeng, "RAS: a robust authentication scheme for filtering false data in wireless sensor networks," in *Proceedings of the 15th IEEE International Conference on Networks (ICON '07)*, pp. 200–205, Adelaide, Australia, November 2007.

[9] T. Yuan, S. Zhang, Y. Zhong, and J. Ma, "KAEF: an en-route scheme of filtering false data in wireless sensor networks," in *Proceedings of the IEEE International Performance Computing and Communications Conference (IPCCC '08)*, pp. 193–200, December 2008.

[10] C. Krauß, M. Schneider, K. Bayarou, and C. Eckert, "STEF: a secure ticket-based en-route filtering scheme for wireless sensor networks," in *Proceedings of the 2nd International Conference on Availability, Reliability and Security (ARES '07)*, pp. 310–317, Vienna, Austria, April 2007.

[11] X. Yang, J. Lin, P. Moulema, W. Yu, X. Fu, and W. Zhao, "A novel en-route filtering scheme against false data injection attacks in cyber-physical networked systems," in *Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems*, pp. 92–101, Macau, China, 2012.

[12] Z. Yu and Y. Guan, "A dynamic en-route filtering scheme for data reporting in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 150–163, 2010.

[13] L. Yu and J. Li, "Grouping-based resilient statistical en-route filtering for sensor networks," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 1782–1790, Rio de Janeiro, Brazil, April 2009.

[14] C.-M. Yu, C.-S. Lu, and S.-Y. Kuo, "Poster abstract: a dos-resilient en-route filtering scheme for sensor networks," in *Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '09)*, pp. 343–344, New Orleans, La, USA, May 2009.

[15] Y.-S. Chen and C.-L. Lei, "Filtering false messages en-route in wireless multi-hop networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '10)*, pp. 1–6, Sydney, Australia, April 2010.

[16] R. Lu, X. Lin, H. Zhu, X. Liang, and X. Shen, "BECAN: a bandwidth-efficient cooperative authentication scheme for filtering injected false data in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 32–43, 2012.

[17] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh, "Toward resilient security in wireless sensor networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '05)*, pp. 34–45, Chicago, Ill, USA, May 2005.

[18] K. Ren, W. Lou, and Y. Zhang, "LEDS: providing location-aware end-to-end data security in wireless sensor networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, pp. 1–12, Barcelona, Spain, April 2006.

[19] H. Yang and S. Lu, "Commutative cipher based en-route filtering in wireless sensor networks," in *Proceedings of the IEEE 60th Vehicular Technology Conference: Wireless Technologies for Global Security*, pp. 1223–1227, Los Angeles, Calif, USA, September 2004.

[20] H. Wang and Q. Li, "Achieving robust message authentication in sensor networks: a public-key based approach," *Wireless Networks*, vol. 16, no. 4, pp. 999–1009, 2010.

[21] C.-M. Yu, C.-S. Lu, and S.-Y. Kuo, "A constrained function based message authentication scheme for sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '09)*, pp. 1–6, Budapest, Hungary, April 2009.

[22] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 247–260, 2006.

[23] C. Krauß, M. Schneider, and C. Eckert, "Defending against false-endorsement-based DoS attacks in wireless sensor networks," in *Proceedings of the 1st ACM Conference on Wireless Network Security (WiSec '08)*, pp. 13–23, New York, NY, USA, April 2008.

[24] J. Deng, R. Han, and S. Mishra, "Defending against path-based dos attacks in wireless sensor networks," in *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '05)*, pp. 89–96, Alexandria, Va, USA, November 2005.

[25] H. Yu and J. He, "Trust-based mutual authentication for bootstrapping in 6LoWPAN," *Journal of Communications*, vol. 7, no. 8, pp. 634–642, 2012.

[26] H. Yu, J. He, T. Zhang, P. Xiao, and Y. Zhang, "Enabling end-to-end secure communication between wireless sensor networks and the internet," *World Wide Web*, vol. 16, no. 4, pp. 515–540, 2013.

[27] X. Cao, W. Kou, X. Zeng, and L. Dang, "Identity-based anonymous remote authentication for value-added services in mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3508–3517, 2009.

[28] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, pp. 427–438, 1987.

[29] W. Lou, W. Liu, and Y. Fang, "SPREAD: enhancing data confidentiality in mobile ad hoc networks," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 2404–2413, Hong Kong, China, March 2004.

[30] J. Lin, X. Yang, W. Yu, and X. Fu, "Towards effective en-route filtering against injected false data in wireless sensor networks," in *Proceedings of the 2011 IEEE Global Telecommunications Conference*, pp. 1–5, Houston, Tex, USA, December 2011.

[31] S. Chapra, *Applied Numerical Methods W/MATLAB: For Engineers & Scientists*, McGraw-Hill Science, New York, NY, USA, 3rd edition, 2011.

[32] Crossbow, "MICA2," http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 842608, 11 pages
http://dx.doi.org/10.1155/2013/842608

Research Article

A Hybrid Security Mechanism for Intra-WBAN and Inter-WBAN Communications

Sarah Irum,[1] Aftab Ali,[1] Farrukh Aslam Khan,[1,2] and Haider Abbas[2,3]

[1] *National University of Computer and Emerging Sciences, Islamabad 44000, Pakistan*
[2] *King Saud University, Riyadh 11653, Saudi Arabia*
[3] *National University of Sciences & Technology, Islamabad 44000, Pakistan*

Correspondence should be addressed to Farrukh Aslam Khan; fakhan@ksu.edu.sa

Received 27 January 2013; Revised 25 May 2013; Accepted 10 June 2013

Academic Editor: Muhammad Khurram Khan

The emerging wireless body area networks (WBANs) have a great potential for the growth and development of future ubiquitous healthcare systems. However, due to the use of unreliable wireless media, WBANs are exposed to a variety of attacks. The prevention of these attacks depends upon the cryptographic techniques. The strength of cryptography is based on the keys used for encryption and decryption in the communication process. Security is still an alarming challenge for WBANs and needs attention of the research community. The proposed work introduces a hybrid key management scheme for both intra-WBAN and inter-WBAN communications. The proposed technique is based on preloaded keys as well as keys automatically generated from biometrics of the human body. The biometric-based calculations are of linear time complexity to cater the strict resource constraints and security requirements of WBANs. The proposed security mechanism provides an efficient solution for the security of both intra-WBAN and inter-WBAN communications. The results of the proposed technique are compared with an existing key management technique known as BARI+. The results show significant improvement over the results produced by BARI+ in terms of storage, communication, energy overhead, and security.

1. Introduction

Wireless body area network (WBAN) is a special type of network in which sensors are deployed on the human body. The sensors collect physiological values from the body and transmit the collected records to the concerned medical server. The applications of WBANs include health monitoring of patients in a hospital and monitoring of soldiers in a battlefield. The WBAN monitoring system is used to monitor a person's vital signs remotely. The system also receives feedback for maintaining a good health status of the subject so that proper action can be taken to rectify the abnormalities [1, 2]. The introduction of WBANs to E-Health monitoring system has revolutionized the field of health monitoring and resulted in better quality of life [3]. Since we deal with the personal information of a person, the security and privacy becomes an essential part of this communication. In case of medical applications, the security threats may lead a patient to a dangerous condition, and sometimes to the death of the patient [4]. WBAN communication can be classified into intra-WBAN communication and inter-WBAN communication. Intra-WBAN communication refers to the on-body sensors communication while inter-WBAN communication refers to the communication between two different WBANs. WBAN communication faces security issues as biomedical sensors implanted on the human body for mobile healthcare monitoring communicate with external networks, which increases the security risk. Since biomedical sensor nodes are allowed to monitor and transmit potentially sensitive medical data, the security and privacy becomes a major concern in WBANs. WBANs consist of lightweight sensors, which are limited both in terms of computational and communication resources; therefore, the security models and protocols used for wireless sensor networks (WSNs) cannot be applied to WBANs in exactly the same manner for different resource-constrained applications and scenarios [5, 6].

The existing key management techniques for WBANs are either plug-and-play or based on preloading. The work proposed in this paper consists of a hybrid technique; that is, it supports both plug-and-play capability as well as some

predeployment of keys in order to strengthen the security in WBANs. In intra-WBAN communication of the proposed technique, the sensors measure physiological values (PVs) of the human body, and then by using those PVs, the keys are calculated among the sensor nodes. All this process is carried out in an automatic manner; that is, the sensors are put on the human body and the keys are calculated automatically for secure communication. In intra-WBAN communication, our technique has linear time complexity that is $O(n)$ for feature generation from electrocardiogram (EKG) signals. The proposed inter-WBAN communication is purely based on preloading of keys. We use minimum number of keys for preloading in the sensor's memory due to its small storage capability. So our hybrid technique is efficient in terms of memory utilization and also in terms of security because the combination of auto key generation and preloading of keys strengthens the security of the technique. Inter-WBAN communication includes the communication between personal servers (PSs). The communication among different PSs is needed when a PS is out of range of a medical server (MS). A PS communicates with another PS and transmits its data to the MS through the nearby PS. Our scheme supports the use of biometric measurements. Keys are generated with the help of biometrics of any PS. The PS generates key pool using its biometric values and then transmits to the whole network. Our scheme also makes use of key refreshment mechanism schedule. MS assigns any PS (key generator) the responsibility of refreshing the key. A list of the notations used in this paper is found in the abbreviations section.

The major contributions of the proposed work are summarized as follows. (1) For intra-WBAN communication, we propose a hybrid scheme by keeping in view the security requirements, storage, and power constraints of a WBAN. (2) For inter-WBAN communication, we propose a lightweight key management scheme based on preloading of keys. (3) The proposed scheme for inter-WBAN communication also uses PVs for the generation of keys in key refreshment phase. (4) The security analysis is done by keeping in view the attacks on both intra-WBAN and inter-WBAN communications. (5) The storage and communication overhead, as well as energy efficiency, are analyzed by comparing the proposed technique with a well-known key management technique known as BARI+ [7].

The rest of the paper is organized as follows. In Section 2, the related work is presented. Section 3 discusses the system model whereas Section 4 describes the proposed technique for intra-WBAN communication. Section 5 describes the proposed technique for inter-WBAN communication. The performance of our proposed technique is analyzed in Section 6, whereas Section 7 concludes the paper.

2. Related Work

The first work that addresses the issue of security for implantable and wearable medical sensors was presented in [8]. These devices are used for nursing human body over long periods of time [9]. Ensuring the security of communication among these devices is critically important [10, 11]. Some works describe the use of human body as a means of generating cryptographic keys for securing intersensor communication. Human body can produce many specific physiological values that are time-variant and are not easy to guess [12]. Using this property of human body for cryptographic purposes provides strong security and gives us great opportunity for automatic key distribution and plug-and-play capability. Both the sender and the receiver can now measure the physiological values from their environment and use them for security purposes whenever they want to communicate [8]. The services like confidentiality and integrity are also ensured in some previous works as discussed in the TLS (transport layer security) protocol [13]. TLS provides privacy and data integrity between two communicating applications. SHELL [14] is a scalable, hierarchical, efficient, and location-aware key management scheme for WSNs. SHELL also provides integrity and confidentiality services in WSNs. SHELL is based on exclusion basis systems (EBSs), which is a combinatorial formulation of the problem of group key management. The main drawback of these protocols for using in WBANs is that these protocols do not fulfill the storage and power limitations of WBANs. Hence, these protocols are not suitable to be used in WBANs. There are some symmetric key management schemes available in the literature for secure trust establishment such as predeployment of keys in nodes, intersensor-communication based key agreement, and public-key-based key agreement schemes. Each of these schemes has its own limitations like memory problem, authentication from a centralized authority, complex mathematics, and so forth, which make these schemes difficult to use in WBANs [8].

Since recently, researchers have been focusing on applications of WBANs and have designed key management techniques for WBANs by using physiological values of the human body such as EKG. The use of EKG signal for generating pairwise keys brings plug-and-play capability in WBANs. Both communicating sensors first sense the EKG values and then, by applying certain hashing and watermarking technique, exchange these values for generating common keys for communication [15, 16]. In [17, 18], the idea of cluster-based secure key agreement protocol for WBANs is presented. The authors use physiological value-based keys for secure cluster topology formation. In [19, 20], the interpulse interval derived from ECG/PPG signals is used to generate common cryptographic keys. In [21], the authors proposed the use of fuzzy vault for physiological signal-based key agreement (PSKA) to secure intersensor communication. In [11], the authors proposed a lightweight security scheme for WBAN communication. The authors also proposed a microcontroller design to reduce energy consumption in WBAN communication. Restrained energy model is considered in this approach where star topology is used for WBAN communication using time division multiple access medium access control (TDMA MAC). The energy overhead is evaluated for the security mechanism introduced in the WBAN. The authors in [22] proposed a security mechanism for WBANs. They reviewed IBE-Lite [23] technique and addressed its limitations such as exposure of master key, partial heath records decryption problem

after rekeying, and lack of adequate privacy provisioning. To overcome these limitations, the authors proposed a scheme that introduced anonymity and unlinkability and offered authorized access of patients' health information. However, they use third party for key generation that itself introduces overhead to the WBAN communication. In [24], the authors proposed a security suite for WBANs. To improve the security of a WBAN, the authors presented techniques such as independent and adaptive management of keys (IAMKeys) for security in WBANs and key management and encryption for securing intersensor communication (KEMESIS). In the proposed schemes, the keys are generated randomly and the security is ensured by eliminating the key exchange between sensor nodes.

In BARI+ [7], the authors proposed a key management scheme purely based on preloading of keys. They use the concept of preloading in intra-WBAN communication. However, in intra-WBAN communication, the advantages of preloading are not so useful due to the fact that preloading-based schemes have no variations and same keys are used for communication between different sensor nodes, whereas in PV-based solution every node that wants to communicate with another node will calculate its own keys. Similarly, in preloading-based schemes, if a key is captured during a communication process, the next key calculations and communications are totally based on that captured key. So, the newly calculated key will also be compromised. While in PV-based key generation, the next set of values cannot be guessed. Preloading of keys for WSNs has also been used in several other papers such as [25–27]. All the PV-based techniques discussed so far are developed purely for intra-WBAN communication without considering inter-WBAN communication. Also, there is a need for a hybrid kind of key agreement scheme that combines the advantages of both PV and preloading-based key agreement. The work presented in this paper uses a hybrid approach for key agreement in WBANs, which tackles the problems of both intra-WBAN and inter-WBAN communications. We present a technique that uses preloading of keys and also generates biometric keys automatically. Keeping in view the strict resource constraints and security requirements of WBANs, minimal preloading of keys is used in the proposed approach.

3. System Model

We assume a WBAN to consist of sensor devices that are capable of measuring biometrics related to human body and also a high power and high storage device known as personal server (PS), which can be a laptop or a hand-held device. Medical server (MS) receives all the information collected by PS through the sensor nodes. All sensor nodes are directly connected to their relevant PSs. Sensor nodes measure biometrics and forward them to the PS. PS in turn transmits collected information to the MS through the internet. Each WBAN is associated with one body. Multiple WBANs are associated with the central MS. PS can communicate with other PSs as well as the MS. The MS stores and processes the information of all the WBANs that are associated with it.

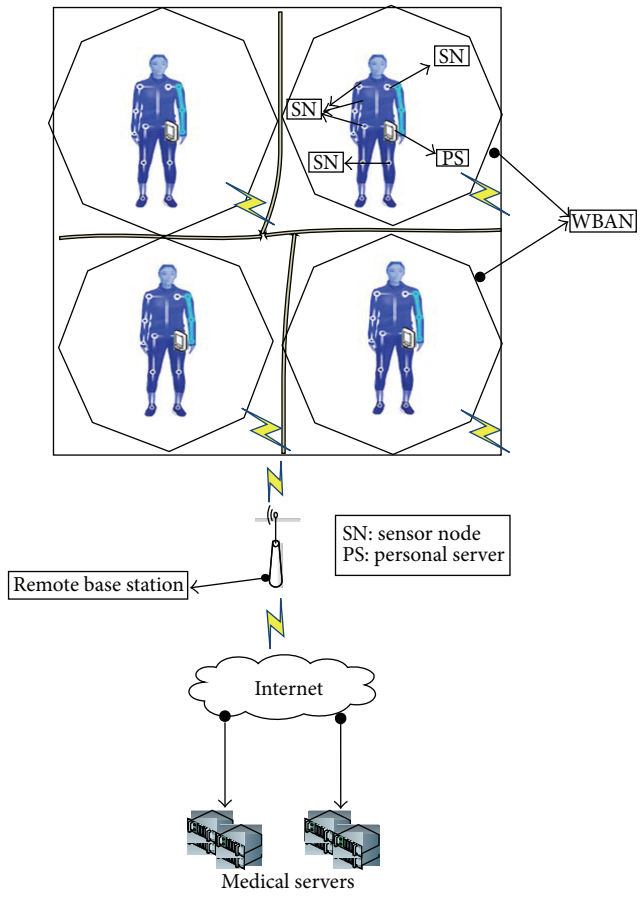


Figure 1: System architecture of wireless body area networks.

All sensor nodes are constrained in energy because they use rechargeable batteries. Sensor nodes are ordinary devices with limited computation, communication, energy supply, and storage capabilities. PS is a powerful node and has more computation, communication, energy supply, and storage capabilities. We assume that the PS is preloaded with node identities and relevant keys before deployment. Keeping in view the storage constraints, in intra-WBAN communication, only one key is preloaded in sensor nodes before deployment. The system architecture of the WBAN, as per our assumptions, is shown in Figure 1. The application scenario of inter-WBAN communication includes multiple bodies under surveillance and all bodies communicate to a remote base station, like in the battlefield, the soldiers are deployed in the enemy territory and they communicate to the remote base station in their own territory. As in Figure 1, the PSs of all the bodies communicate to the base station and then through Internet to the remote MS.

4. Proposed Scheme for Intra-WBAN Communication

Intra-WBAN communication includes the communication of sensor nodes with the PS. We propose a hybrid approach for key management in intra-WBAN communication. Due to the memory limitations, only a single key named as secret

key $K_{SN,MS}$ is preloaded in the sensor nodes and is used in case of PS compromise. Other keys are generated by sensors themselves using their biometrics. The process is done in two steps: feature generation and key agreement.

4.1. Feature Generation. In the feature generation phase, features are extracted and then quantized for secure intersensor communication with the help of EKG using discrete wavelet transform (DWT). DWT allows good localization both in time and spatial frequency domains and is computationally inexpensive. In the process of communication between SNs and PS, sensors sample the EKG signal at the sampling rate of 125 Hz in time duration of 5 seconds. To remove unnecessary frequency components, the signal is then filtered. 625 samples are produced by five-second sample of EKG and then divided into 5 parts of 125 samples each. DWT is applied on each part after applying filtration. The 320 coefficients feature vector is formed by concatenating the 64 coefficients horizontally. In the quantization phase, the generated feature vector is divided into 20 blocks, each containing 16 coefficients, and then they are quantized into a binary stream.

4.2. Key Agreement. After the process of quantization, creation of feature vectors, and formation of blocks, the key agreement process is done. In the key agreement phase, PS broadcasts data request message as shown in message m_1 of Figure 2, consisting of ID_{PS}, DataReq, and nonce. All sensor nodes which have the required data first compute the shared pairwise key with the PS by applying keyed hash function on feature blocks, ID_{PS} and ID_{SN} as follows:

$$\begin{aligned} K_{PS,SN} &= \mathrm{HMAC}\left((b_{11}, N) \cdots (b_{211}, N), \mathrm{ID}_{PS} \parallel \mathrm{ID}_{SN}\right) \\ m_1 &: \mathrm{PS} \longrightarrow * : \mathrm{ID}_{PS}, \mathrm{DataReq}, \mathrm{nonce} \\ m_2 &: \mathrm{SN} \longrightarrow \mathrm{PS} : \mathrm{ID}_{SN}, \mathrm{EK}_{PS,SN}\left(\mathrm{ID}_{SN}, \mathrm{Data}\right), \\ &\quad \mathrm{MAC}_{K_{PS,SN}}\left(\mathrm{ID}_{SN}, \mathrm{nonce}, \mathrm{Data}\right). \end{aligned} \tag{1}$$

SN encrypts the data with key $K_{PS,SN}$ and also computes MAC on ID_{SN}, nonce, and data using the same key $K_{PS,SN}$. SN sends its ID, encrypted data, and MAC to the PS as shown in message m_2. When PS receives this message, first it calculates the $K_{PS,SN}$ by applying the keyed hash function on the feature blocks, ID_{PS} and ID_{SN}. As feature blocks are the same on both sides, $K_{PS,SN}$ generated by PS will be same as that of SN. Incorporation of ID_{SN} in key generation process ensures the establishment of unique pairwise key of PS with all communicating SNs. PS decrypts the message with $K_{PS,SN}$ and compares ID_{SN} and received feature blocks (data) with decrypted message ID_{SN} and feature block on PS to ensure that both parties have generated the same key. The message authenticity is checked by PS through MAC verification with $K_{PS,SN}$.

In Figure 2, the key agreement phase is shown. EKG signal is used for feature generation. Feature vector of 320 coefficients is generated by concatenating 64 coefficients horizontally. These generated features are then divided into 20 blocks of 16 coefficients and then quantized into the binary stream. PS sends data request with its ID to SNs. SNs generate the shared pairwise key with PS by applying keyed hashing on the feature blocks and IDs of both PS and SN. The resulting key is used to encrypt data requested by the PS [9]. Data is verified through MAC verification of the PS. Hamming distance is calculated to verify that the data blocks of both the sender and the receiver are the same.

4.3. Rekeying. In intra-WBAN communication, there is a need to have a common key in order to securely communicate messages to PS. Key is computed after network initialization and generation of the shared pairwise keys between PS and SNs. PS broadcasts a signaling message of GenKey to direct the SNs to generate a common key as follows:

$$m_1 : \mathrm{PS} \longrightarrow * : \mathrm{GenKey}\left(\mathrm{ID}_{PS}\right). \tag{2}$$

Each SN when receives this message generates the key (K) by applying keyed hash function on feature blocks and ID_{PS} as follows:

$$K = \mathrm{HMAC}\left((b_{11}, N) \cdots (b_{211}, N), \mathrm{ID}_{PS}\right). \tag{3}$$

WBAN key K is refreshed after fixed intervals. When PS wants to refresh K, it sends GenKey message and SNs upon receiving this message and regenerates K by applying keyed-hash function on the current feature blocks and ID_{PS}.

5. Proposed Scheme for Inter-WBAN Communication

Inter-WBAN communication includes the communication of a PS with other PSs. Each body in the WBAN contains one PS. The communication of different PSs is needed when a PS is out of range of the MS. PS communicates with other PSs and transmits data to the MS through the nearby PS. Our proposed scheme supports the use of biometric measurements. Keys are generated with the help of biometrics of any PS. The PS generates key pool using its biometrics and then transmits to the whole network. Our scheme for inter-WBAN communication also makes use of key refreshment mechanism schedule. MS assigns any PS (key generator) the responsibility of refreshing the key. Figure 3 shows the manner in which our scheme manages the keys of a WBAN.

Our scheme consists of four types of keys: administrative key (K_{admin}), network key (K_{net}), basic keys of all personal servers, and $K_{MS,PS}$ key shared between MS and PS. Administrative key and basic keys are preloaded in all PSs. Network key K_{net} is a network wide key and is used to transfer data through the network in a secure manner. In our scheme, K_{net} is managed by the MS. Since K_{net} is used very frequently, it may come under cryptanalytic attacks and must be refreshed regularly. Administrative key K_{admin} is used to refresh K_{net}. K_{admin} is also a group key but it is not used as frequently as K_{net}. Naturally, K_{admin} is less exposed as compared to K_{net}. Also, K_{admin} needs to be refreshed through some other key at some point in time. Therefore, we employ basic keys K_{PSbsc} in our key management framework. Every PS has its own K_{PSbsc}, which it shares only with the MS. $K_{MS,PS}$ is used by PS to send data to the MS and it is only shared between PS and MS.

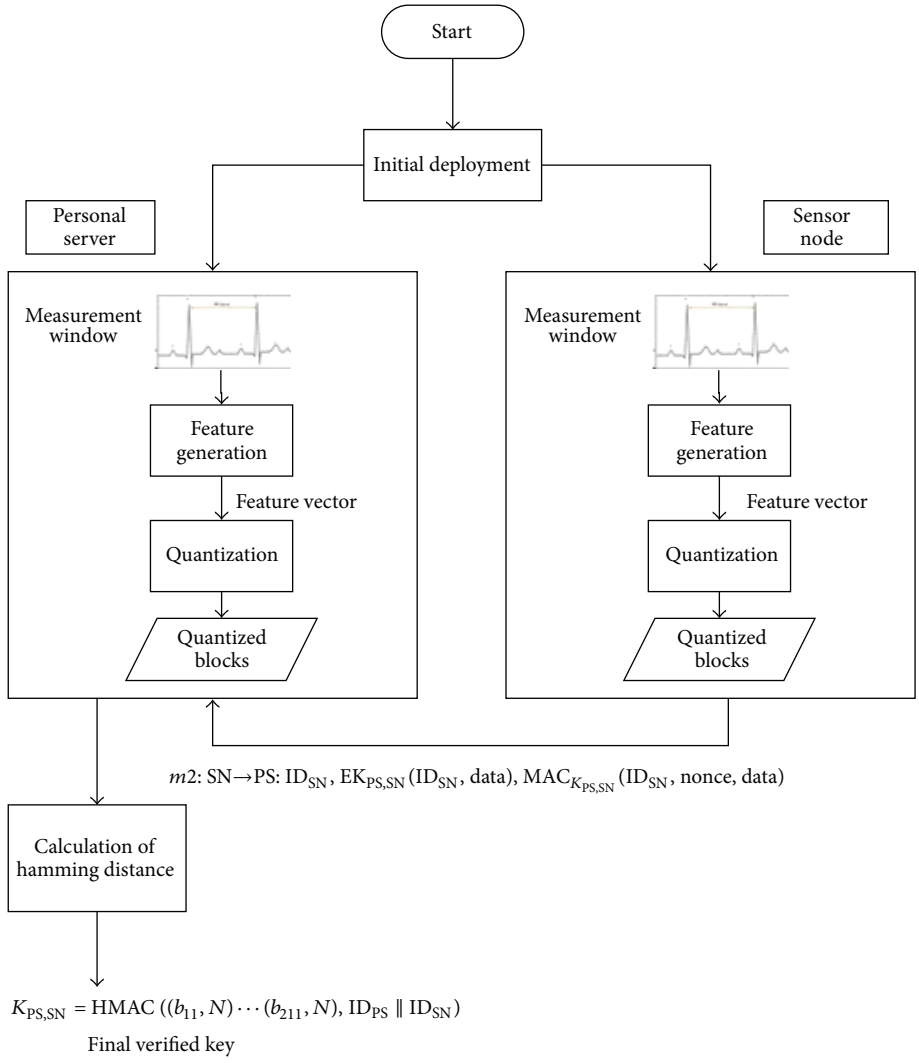


Figure 2: Flowchart of the proposed scheme for intra-WBAN communication.

5.1. Initial Deployment. All PSs are deployed in the beginning. Throughout the network lifetime, PS is connected with the medical server through an external secure communication channel, which may be the Internet. Personal servers come preloaded with K_{admin} and their relevant basic keys and authentication codes. These codes are used to authenticate PSs. After the PS is deployed, sensor devices are deployed on various parts of the body. Soon after deployment, each PS sends discovery message to the MS as follows:

$$m_1 : \forall \text{PS}^i \in \{\text{PS}\} : \text{PS}^i \longrightarrow \text{MS} : \text{EK}^i_{\text{PSbsc}} \left\{\text{ID}^i \mid \text{Auth\_code}^i\right\}. \tag{4}$$

MS authenticates PS and sends the key $K_{\text{MS,PS}}$ and IDs of sensor nodes that are to be deployed in PS:

$$m_2 : \forall \text{PS}^i \in \{\text{PS}\} : \text{MS} \longrightarrow \text{PS}^i : \text{EK}^i_{\text{PSbsc}} \left\{K^i_{\text{MS,PS}} \mid \forall \text{SN}^j \in \{\text{SN}\} : \left\{\text{ID}\left(\text{SN}^i\right)\right\}\right\}. \tag{5}$$

MS assigns any PS the responsibility to generate K_{net} and sends the key refreshment schedule to all PSs in the network as shown in Table 1:

$$m_3 : \text{MS} \longrightarrow \text{KG}^i : \text{EK}_{\text{admin}} \times \left\{\text{Key}_{\text{Gen}_{\text{msg}}} \mid \text{EK}^i_{\text{new}}\text{PSbsc} \mid \text{Timestamp} \mid \text{Key\_Ref\_Schedule}\right\}. \tag{6}$$

The assigned PS generates K_{net} using its biometrics. PS generates key pool with the help of its biometric and assigns K_{net} randomly from its generated key pool:

$$m_4 : \text{KG} \longrightarrow * : \text{EK}_{\text{admin}} \{K_{\text{net}}\}. \tag{7}$$

5.2. Rekeying. In order to refresh K_{net}, MS sends message to KeyGen to refresh K_{net}:

$$m_1 : \text{MS} \longrightarrow \text{KG}^i : \text{EK}_{\text{admin}}$$

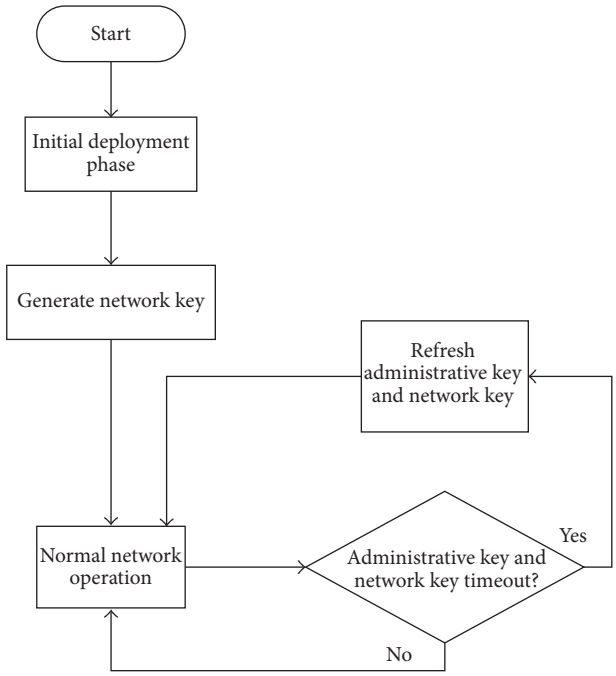


Figure 3: Flowchart of our proposed scheme for inter-WBAN communication.

$$\times \left\{\text{Key\_Ref\_msg} \mid \text{EK}^{i}_{\text{new}}\text{PSbsc} \mid \text{Key\_Ref\_Schedule} \mid \text{Timestamp}\right\}. \tag{8}$$

KeyGen computes new value of K_{net} from its biometrics. It then broadcasts to the whole network encrypting with K_{admin} as follows:

$$m_2 : \text{KG} \longrightarrow * : \text{EK}_{\text{admin}} \left\{K_{\text{new}}\text{net}\right\}. \tag{9}$$

MS sends new refreshment schedule to all PSs encrypting with the current value of K_{admin} when the refreshment schedule expires:

$$m_1 : \text{MS} \longrightarrow * : \text{EK}_{\text{admin}} \left\{\text{Key}_{\text{Ref}_{\text{Schedule}}} \mid \text{Timestamp}\right\}. \tag{10}$$

Administrative key K_{admin} is refreshed periodically. The assigned PS, when its turn arrives computes value from its biometrics. It generates key pool and assigns K_{admin} randomly from the generated values. PS broadcasts newly generated value of K_{admin} to the network:

$$m_2 : \text{KG} \longrightarrow * : E^{K}_{\text{old}}\text{admin} \left\{K_{\text{new}}\text{admin}\right\}. \tag{11}$$

Basic keys K_{bsc} of all PSs are refreshed when they are used.

5.3. Personal Server Addition. We assume that MS contains all the information of the deployed PSs and the newly deployed PSs as all PSs transmit data to the MS. The addition of PS is possible; that is, in case of PS compromise new PS is added to the network. When a PS is added to the network, it sends discovery message to MS as follows:

Table 1: Example of Key Refreshment Schedule with n slots.

New schedule	Turn 1	Turn 2	Turn 3		Turn N
MS	PS_{10}	PS_{5}	PS_{3}	$\cdots$	PS_{6}

$$m_1 : \forall \text{PS}^{i} \in \{\text{newPS}\} : \text{PS}^{i} \longrightarrow \text{MS} : \text{EK}^{i}_{\text{PSbsc}} \left\{\text{ID}^{i} \mid \text{Auth\_code}^{i}\right\}. \tag{12}$$

MS authenticates its ID and authentication code and sends $K_{\text{MS,PS}}$ to the newly deployed PS. MS also sends the information of sensor nodes that are to be deployed in the PS and the relevant keys:

$$m_2 : \forall \text{PS}^{i} \in \{\text{newPS}\} : \text{MS} \longrightarrow \text{PS}^{i} : \text{EK}^{i}_{\text{PSbsc}} \times \left\{K^{i}_{\text{MS,PS}} \mid K_{\text{admin}} \mid K_{\text{net}} \mid \forall \text{SN}^{j} \in \{\text{SN}\} : \left\{\text{ID}\left(\text{SN}^{i}\right)\right\}\right\}. \tag{13}$$

Administrative key and network key are refreshed always when a new PS is added into the network following the same rekeying method.

6. Analysis and Comparison

In this section, we analyze our proposed technique with respect to storage, communication, and energy overhead as well as perform the security and performance analysis. We also compare our proposed technique with a well-known key management technique known as BARI+ [7]. Our proposed scheme involves values that are time variant, that is, EKG values that possess the randomness property. We use HMAC-MD5 for hashing. HMAC-MD5 is more efficient than other hashing techniques as it takes less computation cycles for key generation [28]. Randomness of keys can be determined by calculating the probabilities of keys.

6.1. Storage Overhead. Storage overhead is computed by analyzing all the keys and authentication codes for nodes of different types. Storage requirements of authentication nodes are not included in the analysis. In intra-WBAN communication, we use only one key for communication in the network. Sensor nodes store one key $\text{KS}_{\text{SN,MS}}$ which is computed through biometrics. Two short integers are reserved for the computation of key whereas one short integer is equivalent to 2 bytes. In (14), z is the length of the key:

$$S_{\text{SN}} = z + 4. \tag{14}$$

PS stores all IDs of sensor nodes, $K_{\text{PS,SN}}$ and 4 bytes for the computation of biometric based key. Storing a sensor node's identity requires 2 bytes. Another 2 bytes are required to specify timeout after which the sensor node refreshes $K_{\text{PS,SN}}$. The storage overhead of PS for intra-WBAN communication can be computed using (15). Figure 4 shows the storage

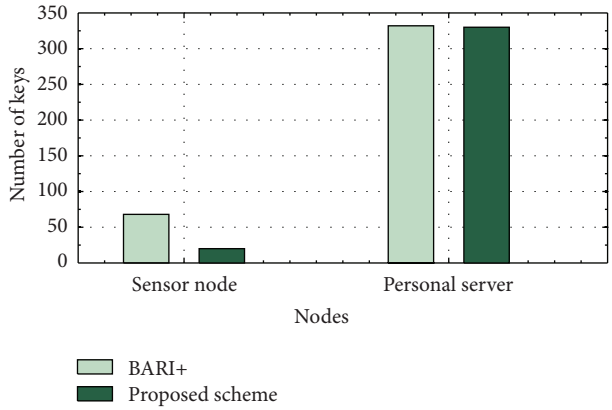


Figure 4: Storage overhead comparison of BARI+ and the proposed scheme for intra-WBAN communication.

Table 2: Storage requirement (in bytes) of each type of nodes for intra-WBAN communication.

Technique	Sensor node	Personal server
BARI+	$(4 \times z) + 4$	$[(r + 2) \times z] + (4 \times r)$
Proposed scheme	$z + 4$	$(r \times z) + (4 \times r) + 4$

overhead of BARI+ and the proposed scheme for intra-WBAN communication. For comparison, the number of sensor nodes (r) is assumed to be 15 and the key size is assumed to be 16 bytes in our simulations. It is evident from Figure 4 that the proposed scheme outperforms BARI+ in case of sensor memory utilization, while in case of PS memory consumption, the proposed scheme is equivalent to BARI+:

$$S_{\text{PS}} = (r \times z) + (4 \times r) + 4. \tag{15}$$

In inter-WBAN communication, only PS takes part in the network communication. So, the PS stores the key refreshment schedule which takes 4 integer bytes. Three keys K_{net}, K_{admin}, and K_{bsc} are stored in PS. PS also stores a key pool of s size and each key of z bytes. Overall storage requirement of PS is calculated as follows:

$$S_{\text{PS}} = (3 \times z) + (s \times z) + 4. \tag{16}$$

Storage requirements of BARI+ and the proposed scheme are shown in Table 2.

6.2. Communication Overhead. Communication overhead is computed for both intra-WBAN and inter-WBAN communication. Intra-WBAN communication overhead is very simple as all nodes are in the range of each other and the average messages transmitted by sensor nodes are very less. For both types of communication, each node sends one message in the initial deployment phase. Table 3 shows the average number of messages transmitted by each type of node in initial deployment phase in both of the schemes.

To refresh K_{net}, MS directs PS to refresh the network key. PS generates new network key by using its biometrics and sends to the whole network. Table 4 shows the average number of messages transmitted by each type of nodes in refreshment of network key in both of the schemes.

Table 3: Average number of messages transmitted by each type of nodes in initial deployment.

Technique	Sensor node	Personal server
BARI+	1	1
Proposed scheme	1	1

Table 4: Average number of messages transmitted by each type of nodes in key refreshment phase.

Technique	Sensor node	Personal server
BARI+	—	1
Proposed scheme	—	2

Table 5: Average number of messages transmitted by each type of nodes when administrative key is refreshed.

Technique	Sensor node	Personal server
BARI+	$1/r$	$((2 \times y) + 1)$
Proposed scheme	—	$1/r$

To refresh K_{admin}, each PS sends one message in every schedule in order to refresh the administrative key. The average messages transmitted by all PSs are $1/r$, if all PSs participate in refreshment of K_{admin}. Table 5 shows the average number of messages transmitted by each type of node in refreshment of the administrative key in both of the schemes.

6.3. Energy Consumption. Energy consumption is computed by calculating the total number of messages transmitted by all types of nodes. Energy is dependent on the distance between the PS and the sensor nodes. As the distance between PS and sensor node increases, its energy consumption also increases. Energy is calculated using the ratio model given in [11, 29, 30]. The following formula is used for the calculation of energy:

$$\text{Energy} = \text{data\_packet} * \left(2 * e\_\text{elect} + e\_\text{emp} * \text{distance}\right). \tag{17}$$

In (17), data_packet represents the number of packets transmitted by all types of nodes, e_elect is the energy consumption in the electronics for sending or receiving one bit, and e_emp is the transmit amplifier. Figure 5 shows the energy consumption in WBAN communication for transmission of 200 data packets. The number of keys used to refresh the administrative key is less than BARI+ that is why energy consumed by BARI+ is very high in the process of administrative key refreshment. In the proposed technique, the messages transmitted to refresh keys are higher as compared to BARI+, that is why more energy is consumed in the key refreshment phase.

6.4. Node Eviction. Node eviction means that any node in the network leaves the network for some reason, for example, power consumption, node emigration, node capture, and so

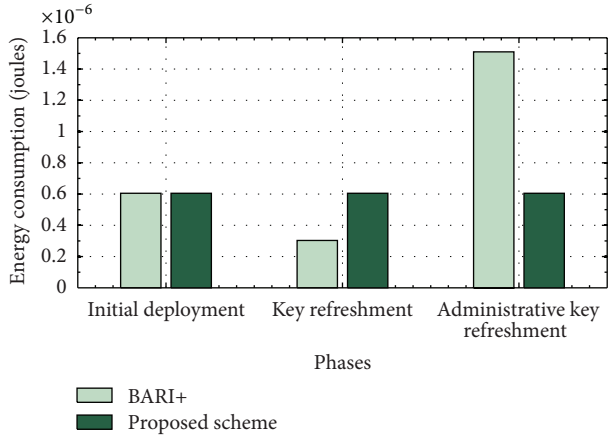


Figure 5: Energy comparison of all phases in BARI+ and the proposed scheme.

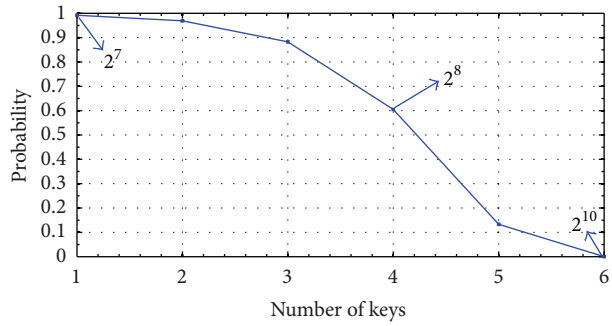


Figure 6: Probability of uniquely generated keys.

forth. We assume that the compromised nodes, the energy-exhausted nodes, and the migrated nodes can eventually be detected by most of its neighbors within a certain time period by sending the keep-alive messages. If a certain node does not respond to a keep-alive message, then the neighboring nodes remove that node from its neighbor list.

6.5. Security Analysis. Our proposed technique is analyzed by considering both insider and outsider attacks. WBAN faces both types of attacks. In passive eavesdropping, the attacker records encrypted keys. In replay attacks, the attacker captures legitimate messages and replays these messages in the network. Insider attacks include physical access of the nodes and attacker can launch multiple attacks such as unauthorized access to data, false injection of data, and alteration of health data.

6.5.1. Outsider Attack. Only the authorized sensors can communicate in the network; that is, without proper authorization, sensor nodes cannot communicate in the network. The communication among the sensor nodes is secured by using the keys like K_{admin}, $K_{\text{MS,PS}}$, K_{bsc}, and K_{net}. Sensors lying outside of the network are categorized as outsiders and cannot participate in the communication without properly assigned key materials. So if an outsider tries to attack, our authentication mechanism provides strong protection against the outsider attacks by ignoring all communications from the stranger nodes.

6.5.2. Replay Attacks. In replay attacks, an attacker stores previous messages and then resends those messages to launch the attack. The proposed scheme uses a nonce and timestamps to prevent the replay attacks. The nonce is checked to see if it duplicates a previously presented value. The timestamp allows receivers to limit how long nonces are retained. If an attacker gains some information and then replays it, the attacker will be caught because of the difference in nonce and timestamp.

6.5.3. PS Compromise. The proposed scheme shows a strong resilience against the capture of the PS. Network key K_{net} is generated using the biometrics of KeyGen. In rekeying technique, the probability of a key repeat in the network depends upon the length of the key and the total number of keys generated by KeyGen. For this purpose, we use the formula of "Birthday Paradox." For 64-bit length key, 264 combinations of keys are used. In "Birthday Paradox" the probability of repeating the key is 0.5 in 232 attempts, which is also a big number. Since a WBAN has much lesser number of nodes than 232 (e.g., 215), the probability of repeating a key in entire network decreases. Due to the randomness property of biometrics, the probability of repeating a key approaches to zero. Figure 6 shows that at initial stage, the probability of keys to be unique remains closer to one. After the initial phase, the curve starts declining and then approaches to zero. The main focus of the analysis is to find a threshold at which the probabilities that all keys generated are unique and no key gets repeated falls within 1 to 0.999999. If a PS is compromised by an adversary, MS revokes the existing keys of the PS. PS is recovered by using the secret key $K_{\text{MS,SN}}$ and the authentication codes of PS are refreshed. In intra-WBAN communication, new PS is verified by the secret key $K_{\text{MS,SN}}$. In inter-WBAN communication, MS directs KeyGen node to generate new Network key K_{net} and Admin key K_{admin} using the rekeying method.

6.5.4. Sensor Node Compromise. The probability of sensor node compromise is less in WBANs as compared to WSNs. However, in case of sensor node compromise, new keys are generated by rekeying method in intra-WBAN communication in the proposed technique.

6.5.5. KeyGen Compromise. In inter-WBAN communication, PS serves as KeyGen node for communication in the network and for rekeying. If a KeyGen node is compromised, the responsibility of generating keys is shifted to another PS by the MS.

6.5.6. Confidentiality. In the proposed technique, the network traffic is secured by encrypting all messages using secret keys. Confidentiality is maintained by protecting data against the unintended parties. An attacker cannot overhear the network unless it obtains the secret key. In the proposed technique, we encrypt the data by using keys. Encrypted data ensures the secure communication of intra-WBAN and inter-WBAN communication.

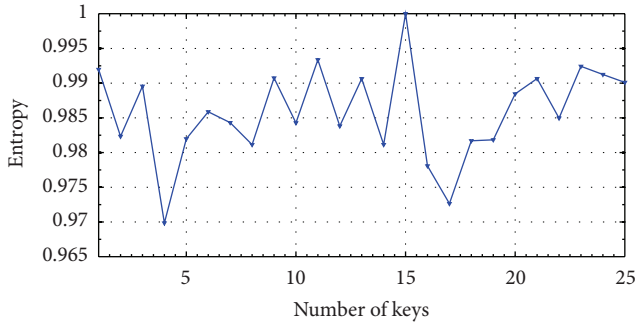


Figure 7: Average entropy of keys for 31 subjects.

6.5.7. Authentication. To protect network from false injection of data, data authentication is required. An attacker can easily inject false message, so the receiver has to make sure that the data received originates from the relevant sender. MAC is applied on each message between PS and sensor node to achieve authentication in the proposed technique for intra-WBAN and inter-WBAN communications.

6.5.8. Integrity. To ensure the integrity of data, we use MAC authentication in intra-WBAN communication in the proposed technique. Alteration and modification of data can be easily determined by using the MAC. Data integrity ensures the accuracy of data being transmitted.

6.5.9. Freshness. The attacker can capture the data and replay it. Data freshness ensures that the frames transmitted are not reused. Data freshness is guaranteed by using rekeying method in both intra-WBAN and inter-WBAN communications.

6.6. Performance Analysis. For the performance analysis of our proposed scheme, we compare different hashing schemes. According to our comparisons by running HMAC-MD5, SHA1 and MD5 for 2.9 seconds, the data is processed by these algorithms against each block size mentioned in Table 6. The results in Table 6 show that HMAC-MD5 processes more bytes of data as compared to other techniques. The implementation is done on a system with 4GB RAM, 2.20 GHZ processor and Red Hat Enterprise Level 5 operating system.

6.6.1. Randomness. For intra-WBAN communication, the randomness of keys is determined by calculating the entropy of the keys using NIST randomness testing suite. Entropy is calculated for 31 subjects over 100 random start times. Entropy of keys almost reaches to 1 in our case, which means that no data is repeated as shown in Figure 7. The purpose of this test is to compare the frequency of overlapping blocks of two consecutive lengths for a random sequence. In the proposed inter-WBAN technique, the randomness of data is ensured by calculating the average entropy of 31 subjects as shown in Figure 7.

Several tests are performed such as frequency, block frequency, cumulative sums, runs, nonoverlapping template, and linear complexity in NIST randomness testing suite as shown in Figure 8. These algorithms are used to test the randomness of the data. By using these algorithms, deviations of a binary sequence from randomness are detected. These tests detect whether the pattern is repeated in the sequence. By these tests, critical value is determined. If the test exceeds the critical value, it means that the data is not random. Results of these tests can be determined by checking the P value of these algorithms. If P value is greater than 0.01, it means that the test is successful and the sequence is random. Simulation is performed on all the above tests for EKG-based data. Figure 8 shows the NIST randomness testing suite results for randomness of the generated keys.

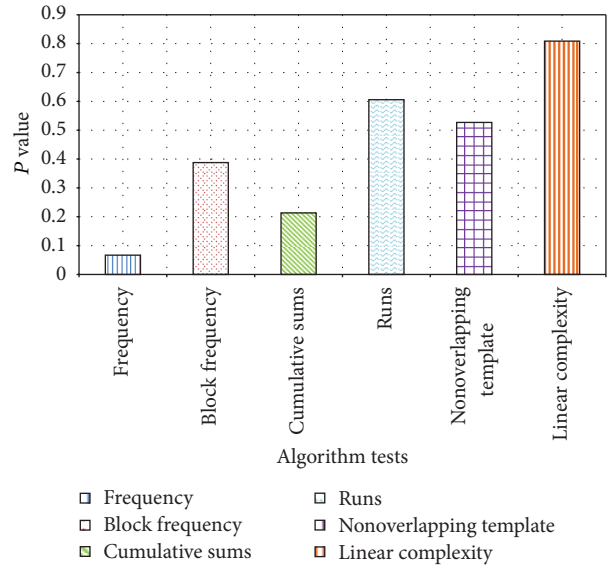


Figure 8: NIST randomness testing suite results for the generated keys.

6.6.2. Distinctiveness. In intra-WBAN communication, distinctiveness or uniqueness of keys is determined by calculating the hamming distance of 31 subject keys. Hamming distance is a measure of calculating the difference between two vectors. Distinctiveness of 31 subject keys means that the keys are identical for the same subject and different for other subjects. The results are shown in Figure 9 at random start time. The figure shows that the values at the diagonal are zero, which means that same subject keys are similar and others are different.

6.6.3. Computational Cost. In the proposed technique, we use only one key for the security of intra-WBAN communication and minimum keys are used for inter-WBAN communication. In intra-WBAN communication, discrete wavelet transform- (DWT-) based solution is used as its computational cost is $O(n)$ [10], which is faster than the techniques such as fast Fourier transform (FFT) used by many other researchers.

7. Conclusion

Wireless body area networks (WBANs) have numerous applications, including patients monitoring and assisted living.

Table 6: Memory usage of different schemes.

Type	Block size				
	16 bytes	64 bytes	256 bytes	1024 bytes	8192 bytes
MD5	15364.25 k	54577.52 k	150091.94 k	272166.53 k	351300.27 k
HMAC (MD5)	22092.06 k	69330.07 k	178271.96 k	292575.70 k	360981.30 k
SHA1	16007.63 k	52851.41 k	135947.13 k	221869.91 k	272486.74 k

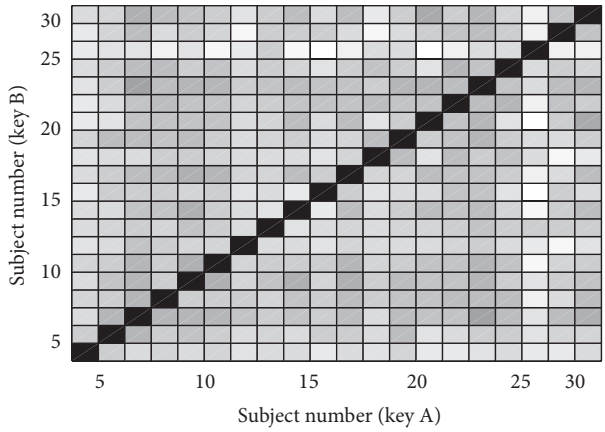


Figure 9: Hamming distance between keys generated from different subjects.

In case of patients monitoring, the human personal data is communicated over an unreliable wireless media, exposing the WBANs to a variety of attacks. Providing a security solution for WBANs will increase the user confidence, which will eventually cause increase in its usability and applicability. The technique presented in this paper is a hybrid security technique for intra-WBAN and inter-WBAN communications. The hybrid technique uses both autogeneration of keys as well as the preloading which makes it efficient in terms of both storage and security. The work presented is twofold; in the first phase, the communication is made secure in intra-WBAN communication by automatically generating keys in sensor nodes and preloading of only one key. In the second phase, the technique is extended to the security of inter-WBAN communication. Security in intra-WBAN is ensured by eliminating key exchange between sensor nodes and the PS. A preloading-based technique is presented for the security of inter-WBAN communication. We analyzed the security, storage requirements, and also its running time by comparing it with an existing technique known as BARI+. The comparison shows that the proposed technique is efficient in terms of all these parameters. Due to its hybrid security mechanism, the technique has a good tradeoff between security and resource constraints.

Abbreviations

WBAN: Wireless body area network
WSN: Wireless sensor network
MS: Medical server
PS: Personal server
SN: Sensor node
$K^i_{SN,MS}$: Key shared between sensor node i and the MS. It is preloaded in every node and refreshed whenever it is used.
K^i_{bsc}: Basic key of PS i shared with the PS. It is preloaded in every node and is refreshed whenever it is used
K_{net}: Network wide key
K_{admin}: Administrative key
m_i: Message number in a particular communication sequence
EK{$A \mid B$}: Values A and B are put together in a block/chunk and then the chunk is encrypted using key K.

Acknowledgment

The authors would like to extend their sincere appreciation to the Deanship of Scientific Research at King Saud University for its funding of this research through the Research Group Project no. RGP-VPP-214.

References

[1] D. Raskovic, T. Martin, and E. Jovanov, "Medical monitoring applications for wearable computing," *Computer Journal*, vol. 47, no. 4, pp. 495–504, 2004.

[2] T. Martin, E. Jovanov, and D. Raskovic, "Issues in wearable computing for medical monitoring applications: a case study of a wearable ECG monitoring device," in *Proceedings of the 4th Intenational Symposium on Wearable Computers*, pp. 43–49, October 2000.

[3] S. Ullah, H. Higgins, B. Braem et al., "A comprehensive survey of wireless body area networks," *Journal of Medical Systems*, vol. 36, pp. 1065–1094, 2012.

[4] S. Saleem, S. Ullah, and H. S. Yoo, "On the security issues in wireless body area networks," *International Journal of Digital Content Technology and Its Applications*, vol. 3, no. 3, 2009.

[5] D. Djenouri, L. Khelladi, and N. Badache, "A survey on security issues in mobile ad hoc and sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 4, pp. 2–28, 2005.

[6] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 8, no. 2, pp. 2–23, 2006.

[7] K.-U. R. S. Muhammad, H. Lee, S. Lee, and Y.-K. Lee, "BARI+: a biometric based distributed key management approach for wireless body area networks," *Sensors*, vol. 10, no. 4, pp. 3911–3933, 2010.

[8] S. Cherukuri, K. Venkatasubramanian, and S. K. S. Gupta, "BioSec: a biometric based approach for securing communication in wireless networks of biosensors implanted in the human body," in *Proceedings of the International Conference on Parallel Processing Workshops (WiSPr '03)*, Taiwan, 2003.

[9] A. Darwish and A. E. Hassanien, "Wearable and implantable wireless sensor network solutions for healthcare monitoring," *Sensors*, vol. 11, no. 6, pp. 5561–5595, 2011.

[10] P. Kumar and H.-J. Lee, "Security issues in healthcare applications using wireless medical sensor networks: a survey," *Sensors*, vol. 12, no. 1, pp. 55–91, 2012.

[11] G. Selimis, L. Huang, F. Massé et al., "A lightweight security scheme for wireless body area networks: design, energy evaluation and proposed microprocessor design," *Journal of Medical Systems*, vol. 35, no. 5, pp. 1289–1298, 2011.

[12] M. Mana, M. Feham, and B. A. Bensaber, "Trust key management scheme for wireless body area networks," *International Journal of Network Security*, vol. 12, no. 2, pp. 75–83, 2011.

[13] T. Dierks and C. Allen, "The TLS Protocol Version 1.0," The Internet Society: Reston, Va, USA, 1999.

[14] M. F. Younis, K. Ghumman, and M. Eltoweissy, "Location-aware combinatorial key management scheme for clustered sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 8, pp. 865–882, 2006.

[15] A. Ali and F. A. Khan, "An improved EKG-based key agreement scheme for body area networks," *Communications in Computer and Information Science*, vol. 76, pp. 298–308, 2010.

[16] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "EKG-based key agreement in body sensor networks," in *Proceedings of the IEEE INFOCOM Workshops*, New York, NY, USA, April 2008.

[17] K. K. Venkatasubramanian and S. K. S. Gupta, "Security for pervasive health monitoring sensor applications," in *Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing (ICISIP '06)*, pp. 197–202, December 2006.

[18] A. Ali, S. Irum, F. Kausar, and F. A. Khan, "A cluster-based key agreement scheme using keyed hashing for Body Area Networks," *Multimedia Tools and Applications*, vol. 66, no. 2, pp. 201–214, 2013.

[19] C. C. Y. Poon, Y.-T. Zhang, and S.-D. Bao, "A novel biometrics method to secure wireless body area sensor networks for telemedicine and M-health," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 73–81, 2006.

[20] S.-D. Bao, Y.-T. Zhang, and L.-F. Shen, "Physiological signal based entity authentication for body area sensor networks and mobile healthcare systems," in *Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society (IEEE-EMBS '05)*, pp. 2455–2458, September 2005.

[21] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "PSKA: usable and secure key agreement scheme for body area networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 1, pp. 60–68, 2010.

[22] C. Huang, H. Lee, and D. H. Lee, "A privacy-strengthened scheme for E-Healthcare monitoring system," *Journal of Medical Systems*, vol. 36, no. 5, pp. 2959–2971, 2012.

[23] C. C. Tan, H. Wang, S. Zhong, and Q. Li, "IBE-lite: a lightweight identity-based cryptography for body sensor networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 6, pp. 926–932, 2009.

[24] R. V. Sampangi, S. Dey, S. R. Urs, and S. Sampalli, "A security suite for wireless body area networks," *International Journal of Network Security & Its Applications*, vol. 4, no. 1, 2012.

[25] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[26] G. Jolly, M. C. Kuscu, P. Kokate, and M. Younis, "A low-energy key management protocol for wireless sensor networks," in *Proceedings of the 8th IEEE International Symposium on Computers and Communication*, 2003.

[27] M. Boujelben, O. Cheikhrouhou, M. Abid, and H. Youssef, "Establishing pairwise keys in heterogeneous two-tiered wireless sensor networks," in *Proceedings of the 3rd International Conference on Sensor Technologies and Applications (SENSORCOMM '09)*, pp. 442–448, June 2009.

[28] F. Kausar, S. Hussain, L. T. Yang, and A. Masood, *Scalable and Efficient Key Management For Heterogeneous Sensor Networks*, Springer Science Business Media, LLC, 2008.

[29] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[30] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication algorithm for wireless microsensor networks," in *Proceeding of the 33rd International Conference on System Sciences*, pp. 1–10.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 740392, 13 pages
http://dx.doi.org/10.1155/2013/740392

Research Article

Integrated Protocols to Ensure Security Services in Wireless Sensor Networks

Mohammed Faisal, Jalal Al-Muhtadi, and Abdullah Al-Dhelaan

Department of Computer Science, King Saud University, P.O. Box 51178, Riyadh 11543, Saudi Arabia

Correspondence should be addressed to Mohammed Faisal; mfaisal@ksu.edu.sa

Received 19 January 2013; Revised 31 March 2013; Accepted 9 April 2013

Academic Editor: Muhammad Khurram Khan

Security in current/next generation wireless sensor network (WSN) is challenging, due to its special characteristics and the scarcity of energy, and processing power. Thus, many security methods are trying to solve the problem of security in wireless sensor networks. Many of these methods use symmetric cryptography, and others use asymmetric cryptography. Security in WSN demands authentication, confidentiality, integrity, balancing the energy cost, nonreputation, and scalability services. In this paper, we introduce two protocols to address these main security services. The proposed protocols integrate public key and symmetric key algorithms to ensure optimal usage of sensors' energy and processing power to provide adequate security in next generation WSN. The proposed protocols utilize Elliptic Curve Cryptography (ECC), Advanced Encryption Standard (AES), Certification Authority (CA), and Registration Authority (RA) to achieve a reasonable tradeoff.

1. Introduction

WSNs are infrastructureless and fully distributed systems of self-configurable and self-organizing. WSNs are used in industry process control, military sensing and tracking, environmental monitoring, patient monitoring, and so forth [1]. In the last two decades, many research efforts proposed various security services for WSNs [2–6]. These works used symmetric or asymmetric cryptography along with other tools to handle security needs for WSNs [7, 8]. In this paper, we integrate the symmetric and asymmetric key cryptography to ensure security and optimize energy usage. The proposed protocols focus on creating a Light Public Key Infrastructure protocol and Special Cluster based Multipath Routing Protocol. The proposed protocols are Light Public Key Infrastructure (L-PKI) protocol and Secure and Energy-efficient Cluster based Multipath Routing (SECMRP) protocol.

Our main contribution is integrating a lightweight PKI that utilizes ECC and AES crypto algorithms with a secure data transmission service via multipath in an efficient manner, in terms of resource and power consumption. The proposed protocols enhance the security of WSN by providing mutual authentication between neighbor nodes and sending the data via multipath for redundancy.

This paper is organized as follows. Section 2 talks about related work. Section 3 explains the WSN architecture. Section 4 gives an overview of the proposed protocols. Section 5 presents the L-PKI protocol. Section 6 presents the SECMRP protocol. Section 7 describes system's experimentation and results. Section 8 compares results between our work and related work. Finally, Section 9 concludes.

2. Related Work

This work combines key management and route discovery protocols in WSN. Thus, related work can be divided into two major subsections: key management protocols and route discovery protocols.

2.1. Key Management Protocols. Using public key cryptography, we can ensure confidentiality, authentication, and integrity. A PKI is an efficient tool for achieving key management in networks [9]. PKI is greedy in terms of resources. It typically consumes a lot of energy and bandwidth through extensive use of public key crypto. Thus, using PKI in WSN is challenging. Many studies focuses on developing public key cryptographic algorithms that are more efficient energy efficient such as Elliptic Curve Cryptography (ECC) [10]. Several

research groups have successfully implemented public key cryptography in WSNs [4–6].

TinyPK attempts to secure sensor networks with public-key cryptography and implements a public key-based protocol that allows authentication and key exchange between an external party and a sensor network [11]. TinyPK focuses on supporting confidentiality and source authentication for sensor network traffic. TinyPK is based on RSA 1024 bits [12]. TinyPK is implemented on UC Berkeley MICA2 motes using the TinyOS development environment. TinyPK uses other symmetric encryption and Diffie-Hellman key agreement algorithm [13] to exchange secret keys. TinyPK uses a CA whose public key has to be preloaded into the nodes during preconfiguration phase. In our approach we use ECC instead of RSA, which would give an advantage for performance and battery consumption. Also, we avoid overusing Diffie-Hellman key exchange to avoid potential man-in-the-middle attacks.

μPKI [9] is a lightweight implementation of PKI for WSNs. In this protocol, two handshakes are used: the first handshake between the BS and each sensor in the network and the second handshake between each pair of node in the network intended to secure sensor to sensor communication. In μPKI, only the BS needs to be authenticated. μPKI uses the public key cryptography in the key distribution operation. μPKI assumes the existence of an offline communication protocol to distribute the public key of the BS to each sensor in the network. Thus, μPKI uses the public key in the handshake between the BS and sensors. In this handshake, each sensor generates and encrypts the session key using the public key of the BS. The purpose of the first handshake is to create a secure end-to-end transmission between each node and the BS. The second handshake, which is between each pair of node, is used to establish a secure channel between them. In this handshake, one of these nodes sends request to the BS. This request contains the identifier of the corresponding node. The BS responds to this request by generating a random key then encrypts a copy for each sensor using the corresponding session key, which has been created in the first handshake. Actually, in μPKI protocol the security can be broken by knowing the public key of the BS. Any malicious node can then encrypt a session key using the BS's public key. The BS plays the main role in the operation of creating the session key between each pair of nodes. Thus, a lot of traffic exists.

2.2. Route Discovery Protocols. Routing in WSNs is challenging, as there are many routing protocols: some producing a single path only, while others produce multipath. To ensure the security of WSNs, the scheme should ensure the security of route discovery and data transmission. Several protocols have been proposed for route discovery.

Secure and Energy-Efficient Multipath Routing Protocol (SEEM) [14] proposed multipath routing protocols where instead of using the initial lowest energy route for communication, the BS finds multipath to the source of the data and selects one to use during the communication. Furthermore, the BS updates the available energy of each node along the path depending on the amount of packets being sent and received [14]. The BS then uses the updated energy state of each node to select new paths. BS in SEEM works as a server, which floods the query to the network. The node that satisfies the query will send a request to the BS for sending a path. SEEM does not use any cryptographic mechanisms to address confidentiality. SEEM constructs disjoint and braided paths using a modification of the Breadth First Search algorithm [14].

INtrusion-tolerant routing protocol for wireless SEnsor NetworkS (INSENS) proposes a multipath routing protocol that minimizes the computation, communication, storage, and bandwidth required at the sensor nodes in the operation of route discovery [15]. INSENS does not rely on detecting intrusions but rather tolerates intrusions by bypassing the malicious nodes. An important property of INSENS is that while a malicious node may be able to compromise a small number of nodes in its vicinity, it cannot cause widespread damage in the network [15]. In INSENS, the BS plays an important role in the operation of routing discovery. The BS always receives knowledge of the topology of the network and finds multipath to each node in the network, and then the BS unicasts the multipath table to each corresponding node [15].

SECMRP uses the concept of multipath and clustering to deal with security and efficiency. SECMRP enhances the security issue of SEEM and, at the same time, uses two disjointed paths to send sensed data to the CH. In SEEM, the BS works as a server, which floods the query to the network, and the node which satisfies the query will send a request to the BS for a path, but CH in SECMRP periodically sends paths to each node. SEEM increases the network lifetime about 35% as compared to directed diffusion. So that we can say that SECMRP is energy efficient and is able to enhance network lifetime due to the roles of CH.

Early preliminary motivation and design for this work were published in [8]. In this paper, we revise the design, provide full implementation, and compare results with related work.

3. The Proposed WSN Architecture

In order to ensure the security services, we divide the sensor network into many clusters and assume the existence of three types of nodes (Figure 1 illustrates the proposed architecture):

(1) sensing nodes;

(2) cluster heads (CH) (one or more);

(3) base stations (BS) (One or more).

Table 1 includes the tasks of WSNs components.

4. Proposed Protocols

The proposed protocols are designed to ensure authentication, confidentiality, balancing the energy cost, nonrepudiation, scalability, and integrity in WSNS. This is done by proposing two protocols, Light Public Key Infrastructure L-PKI protocol and Secure and Energy-efficient Cluster based Multipath Routing SECMRP.

Table 1: Tasks of WSN components.

Member	Explanation
Base station	(i) Collects and processes information from CH. (ii) Assigns identifier ID to both CHs and ordinary nodes. (iii) Determines the CH of each node in the network by assigning the CH's ID and public key to each node, so each node knows its cluster, and each cluster knows its nodes.
CH	CA roles (i) Issues and delivers the digital certificate of both CHs and ordinary nodes "offline." (ii) Signs the digital certificate "offline." RA roles Binds the digital certificates and ID of both CHs and ordinary nodes. Other roles (i) Ensures the security of proposed WSNs via (a) using its public key for authentication operation of each node and its neighbors (in all nodes that belong to the same cluster), (b) validating the certification of node if the node belongs to other cluster via cross-authentication. (ii) Aggregates the sensed information. (iii) Sends the sensed information to the BS. (iv) Acts as a gateway between the sensed region and the BS.
Ordinary nodes	Sense or monitor (according to the application requirement) information and send it to the BS via secure disjoint multipath through the CHs.

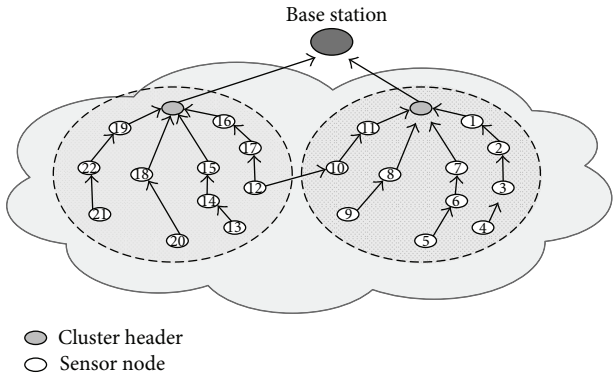


Figure 1: The proposed architecture.

The proposed protocols (L-PKI and SECMRP) prevent from several types of attacks, internal, passive, and impersonation (spoofing) attacks. Also, modification of protocol messages and routing table overflow attacks are addressed in the proposed protocols.

5. L-PKI Protocol

L-PKI proposes a lightweight implementation of PKI, suitable for WSNs. L-PKI proposes a protocol that utilizes public key cryptography only in authentication and in generating a session key between the cluster head (CH) and each sensor node in the cluster. L-PKI sets up secured links between each node and its neighbors, each node and its CH, and each CH and the BS. These secured links are used to transmit data to the BS via CHs.

For L-PKI, we assume the following:

(1) CHs have more computational and energy power compared to other sensor nodes, which allow them to work as a limited Certification Authority (CA) and Registration Authority (RA);

(2) the BS and CHs are trusted entities;

(3) each sensor node has the capability to use symmetric encryption and Elliptic Curve Cryptography (ECC) operations (hardware or software).

It is established that ECC is a relatively efficient public key crypto, relevant to RSA and other popular algorithms, thus it is suitable for WSNs [4–6, 16, 17]. L-PKI utilizes ECC with keys of 160 bits [18]. There are many symbols used in this protocol. To simplify the description of the L-PKI protocol, Table 2 explains the meaning of the symbols used in the protocol description. L-PKI is composed of two phases as follows:

(1) initial authentication and Key establishment phase, which contains

(a) predeployment step,

(b) authentication and key establishment step,

(2) addition of new node phase.

5.1. Initial Authentication and Key Establishment Phase

5.1.1. Predeployment Step. In order to increase the live time of the WSNs, L-PKI moves most of the load of the PKI operations to the CHs. According to the proposed protocol, the CHs will work as a lightweight PKI. Predeployment step includes the following:

(1) each CH and node generate its private and public keys (CHK_{pub}, CHK_{prv}), (NK_{pub}, NK_{prv});

(2) BS generates a unique identifier, CH-ID (8 bits) to each CH in the network. 8 bits CH ID can have up to 256 clusters;

Table 2: Output of LPKI.

Symbols	Description	Size
CH-ID	Identify definition of the CH	8 bits
N-ID	Identify definition of sensor node (N can be any letter according to the name of node)	8 bits
CHK_{pub}/CHK_{prv}	The private and public keys of CH	160 bits
NK_{pub}/NK_{prv}	The private and public keys of sensor nodes (N can be any letter according to the name of node)	160 bits
CH_cer	Digital certificate of CH	192 bits
N_cer	Digital certificate of sensor node (N can be any letter according to the name of node)	192 bits
CH_sig	The signature of CH (N can be any letter according to the name of node)	320 bits
N_sig	The signature of node (N can be any letter according to the name of node)	320 bits
SH_XYK	AES shared key between neighbors node (X and Y can be any letter according to the name of nodes)	128 bits
NBR_LIST	List contains all neighbors of each node	

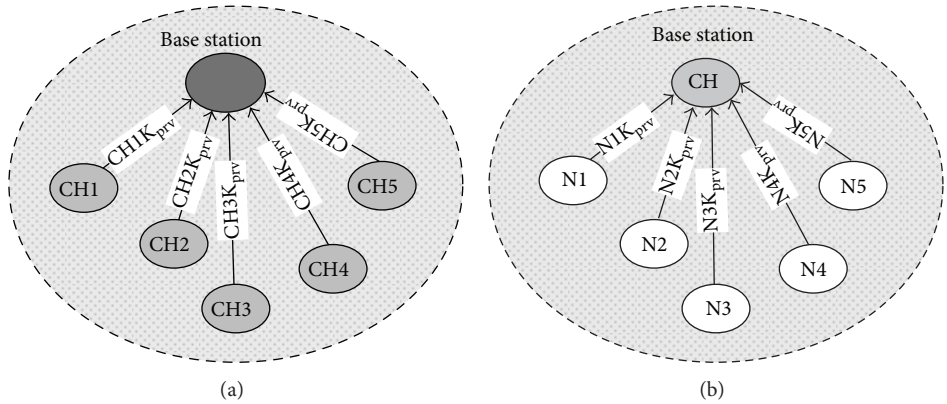


Figure 2: (a) Delivering CHs' key and (b) delivering nodes' key.

(3) BS generates a unique ID (N-ID) (8 bits) to each sensor node in the network;

(4) each CH delivers its public key to BS, as illustrated in Figure 2(a);

(5) each node delivers its public key to its CH. As illustrated in Figure 2(b);

(6) BS issues digital certificates for each CH (CH_cer) and signs it;

(7) BS delivers the digital certificates to each corresponding CH;

(8) CHs issue digital certificates for each sensor nodes (N_cer) and sign it. Figure 3(a) illustrates this step;

(9) CHs deliver the digital certificates to each corresponding node. Figure 3(b) illustrates this step.

L-PKI utilizes a compressed certificate format, which only keep needed fields, (mainly, nodes or CHs IDs, timestamp, and signature). L-PKI uses these fields in order to ensure the security with lowest cost as follows:

(a) CH_cer = CHK_{pub} "160 bits" | CH-ID "8 bits" | Timestamp "16 bits" | BS-ID "8 bits" = 192 bits,

(b) N_cer = NK_{pub} "160 bits" | N-ID "8 bits", Timestamp "16 bits" | CH-ID "8 bits" = 192 bits.

We recommend using Elliptic Curve Digital Signature Algorithm (ECDSA-160) as a signature algorithm [16, 18], so that the size of the signature will be 320 bits, because we use ECC-160. Figure 4 illustrates thesignature operation.

After applying the predeployment step, each node has its own digital certificate N_cer, public key NK_{pub}, and private key NK_{prv} and also knows the public key (CHK_{pub}) and the CH-ID of its CH.

5.1.2. Authentication and Key Establishment Step. At this step, each node in the same cluster must authenticate each other. Figure 5 illustrates the authenticate operation. After the authentication step, each node will trust its neighbor. After the predeployment step, each node will have its own digital certificate N_cer, public key NK_{pub}, private key NK_{prv}, the public key (CHK_{pub}) of its CH, and the CH-ID of its CH. Each node must do the following steps in order to authenticate with its neighbors:

(1) node "A" sends its digital certificate A_cer to its neighbor "B";

(2) node B receives the certificate of node A "A_cer" and verifies it by using the public key of its CH CHK_{pub} as follows:

(i) Size of message = 192 bits "A_cer" = 192 bits;

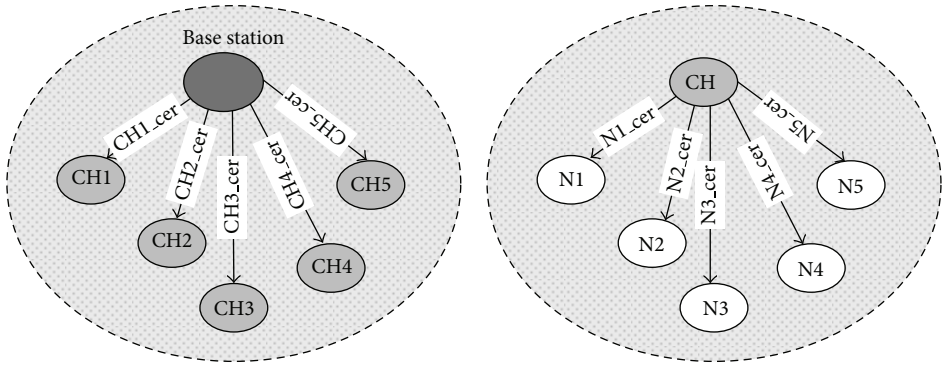


Figure 3: (a) Delivering CHs' digital and (b) delivering nodes' digital certificates.

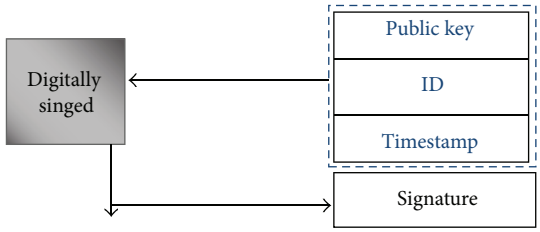


Figure 4: Signature operation.

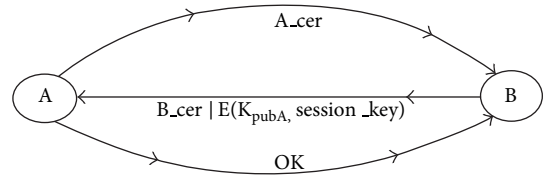


Figure 5: Authentication operation.

(3) at node B, if A_cer is valid, node B creates and encrypts a session key SH_ABK. This key is encrypted by using the public key of node A "AK_{pub}" as follows:

(i) L-PKI uses AES-128, and thus the key size is 128 bits;

(ii) size of message = key 128 bits "SH_ABK" + 192 bits "A_cer" + 128 bits MAC = 448 bits. Figure 5 illustrates the authentication operation;

(4) node B sends its digital certificate B_cer with its session key "SH_ABK" and MAC;

(5) node A receives the certificate of node B "B_cer" and verifies it by using the public key of the CH CH_{Kpub};

(6) if B_cer is valid, node A decrypts SH_ABK by using its private key AK_{prv};

(7) create MAC and compare it with the received MAC;

(8) node A responds by sending OK message with timestamp to node B. The OK message is encrypted by the session key SH_ABK. This message will act as a challenge message between nodes A and B;

(9) if both nodes A and B successfully validate the certificate,

(i) node A adds node B to its neighbor list,

(ii) node B adds node A to its neighbor list.

Now, each node trusts its neighbor node.

5.2. Adding New Node Phase. In case of a new node joining the network, it must be prepared by the BS and CH; that is, it must follow the predeployment step and authentication and key establishment step. At first, the new node has to create NK_{pub}, NK_{prv} and loads the public key of the CH. The BS must assign CH-ID and assign node identifier N-ID. Then, the node must communicate with its CH to prepare its certificate N_cer and determine the cluster of the node. After this, the new node is deployed to the network. The location of the new node is not relevant here, because it can authenticate with any node even a node from another cluster. In the authentication step, if the node belongs to another cluster, the other node sends the node's certificate to the CH. Thus, the CH verifies the certificate by using the public key of the CH which the node belongs to. Figure 6 shows this situation. After the authentication, the new node can establish a new session key.

6. SECMRP Protocol

SECMRP enhances the security issue of SEEM [14] and at the same time uses disjointed paths to send data to the CH. In SEEM, the BS works as a server, which floods the query to the network, and the node which satisfies the query will send a request to the BS for a path. In SECMRP, each CH collects a neighbor table of each node in its cluster then creates and delivers two disjointed paths from CH to each node. SECMRP assumes that each node knows its neighbors, has a shared key with each neighbor and a shared key with its CH. This is done after applying L-PKI protocol. After L-PKI, each node has a unique ID N_ID, a digital certificate N_cer signed by its CH, list of all authenticated neighbors NBR_LIST, public keys of CH CHK_{pub}, and shared key with each

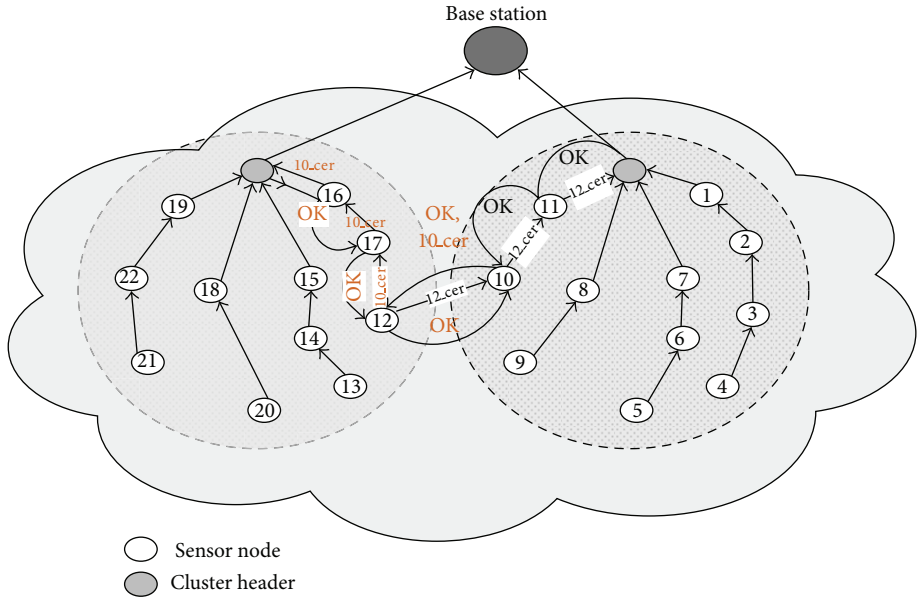


Figure 6: Adding a new node.

Table 3: Symbols of SECMRP.

Symbols	Description	Bits
CH-ID	Identify definition of the CH	8
N-ID	Identify definition of sensor node (N can be any letter according to the name of node)	8
CHK_{pub}/CHK_{prv}	The private and public keys of CHs	160
NK_{pub}/NK_{prv}	The private and public keys of sensor nodes (N can be any letter according to the name of node)	160
CH_cer	Digital certificate of CH	192
N_cer	Digital certificate of sensor node (N can be any letter according to the name of node)	192
CH_sig	The signature of CH (N can be any letter according to the name of node)	320
N_sig	The signature of node (N can be any letter according to the name of node)	320
SH_XYK	AES shared key with each neighbor node (X and Y can be any letter according to the name of nodes)	128
SH_NCHK	Shared key with the CH (N can be any letter according to the name of node)	128
Received_pkts	Table contains the receiving packet sequence number in each node	
P_SEQ_NUM	Packet sequence number	8
N_MAC_val	MAC values of node N (N can be any letter according to the name of node)	128 bits
A_*NBR*	The neighbors of node A	Varies (32 per neighbor)
NBR_SEN	Packet from the CH to ask each node to send its neighbors	160 bits
NBR_LIST	Packet contains all neighbors of node	$424 + n * 32\ (n = 4) \rightarrow = 552$
DATA_PACKET		320 + Size (data)

neighbor node SH_XYK. Table 3 describes all symbols used in SECMRP.

SECMRP has three phases *secure route discovery*, *secure data transmission*, and *route maintenance*. Secure route discovery phase is responsible for finding secure disjoint multipath or partially disjoint multipath between each node and its CH. Data transmission phase is responsible for transmitting the sensing data. At route maintenance phase, each CH updates available energy of each node, which is participating in the communication and according to the available energy on each node reselects a new path to the source node.

6.1. Secure Route Discovery. Secure route discovery phase starts by applying the L-PKI. Secure route discovery has three steps: NBR_LIST requesting, NBR_LIST sending, and NBR_LIST receiving and paths creation steps.

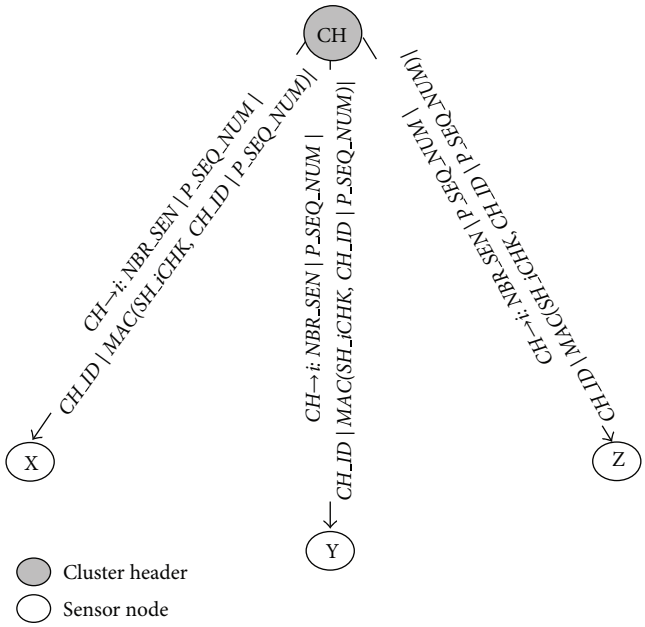


Figure 7: CH requests neighbor list of each node.

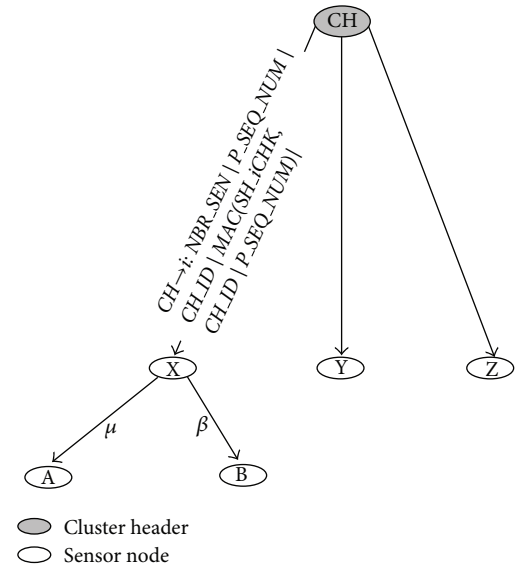


Figure 8: CH requests neighbor list of each node.

```
packet NBR_SEN_Packet
        {
string Packet_Type;
string P_SEQ_NUM;
int CHID;
string Packet_MAC;
        };
```

Data Structure 1: NBR_SEN data type.

6.1.1. NBR_LIST Requesting. Each CH of each cluster starts the routing discovery operation by sending NBR_SEN "neighbor sending" packet to each node on its cluster. The size of NBR_SEN packet is 160 bits and contains packet sequence number "*P_SEQ_NUM*" CH ID "*CH_ID*", and MAC values of the *CH_ID* and *P_SEQ_NUM* "*CH_MAC_val*" the following format (as illustrated in Figure 7.)

for all i where i is a neighbor of CH
CH → *i: NBR_SEN* | *P_SEQ_NUM* | *CH_ID* | *MAC* (*SH_CHiK, CH_ID* | *P_SEQ_NUM*).

Each node receiving NBR_SEN packet does the following:

(1) checks if the node has received this NBR_SEN by searching *P_SEQ_NUM* in the table received_pkts. If the packet received once, then drops this packet and does not rebroadcast it. Otherwise stores *P_SEQ_NUM* in received_pkts table;

(2) computes the MAC value of *CH_ID* and *P_SEQ_NUM* by using the shared key SH_CHiK MAC (SH_CH*i*K, *CH_ID* | *P_SEQ_NUM*) and compares it with the MAC value *CH_MAC_val* of the NBR_SEN packet. If the MAC values are equal move to step 4;

(3) computes the MAC value of CH_ID and *P_SEQ_NUM* by using the shared key of each neighbor node,

for all j where j is a neighbor of i,
MAC (*SH_ijK,* CH_ID | *P_SEQ_NUM*);

(4) rebroadcasts the NBR_SEN packet to its neighbors,

for all j where j is a neighbor of i
i → *j: NBR_SEN* | *P_SEQ_NUM* | *CH_ID* | *MAC* (*SH_ijK, CH_ID* | *P_SEQ_NUM*).

The first step ensures that no NBR_LIST packet is broadcasted more than one time for each node. The second step ensures the integrity and avoids replay attaches. Thus the communication overheads for transmitting and receiving packets are reduced. The following packets and Figure 8 show this operation (X is the current node, and A, B are its neighbors) as follows:

X → ***A***: *NBR_SEN* | *P_SEQ_NUM* | *CH_ID* | *MAC* (**SH_XAK**, *CH_ID* | *P_SEQ_NUM*) *(μ).*

X → ***B***: *NBR_SEN* | *P_SE Q_NUM* | *CH_ID* | *MAC* (**SH_XBK**, *CH_ID* | *P_SEQ_NUM*) *(β).*

The format "data type" of the NBR_SEN packet is shown in Data Structure 1. Each node which has received the NBR_SEN packet responds by sending a neighbor list (NBR_LIST) packet to the CH.

6.1.2. NBR_LIST Sending. After each node has received the NBR_SEN packet and completed all operations that belong to the NBR_SEN packet, it becomes able to send the NBR_LIST packet. Each node prepares the NBR_LIST packet and sends it directly to the node, which has already sent the NBR_SEN packet as follows:

for all i where i is any node in the cluster, j is the node, which has sent the NBR_SEN packet to the node i.

```
packet NBR_LIST_Packet
  {
string Packet_Type;
string P_SEQ_NUM;
int Node_ID;
string Node_Name;
string Packet_MAC_Neighbors; //Mac (SH_key, P_SEQ_NUM|ID)
string Encrypted_Session_Key; //use to save the E (CHK_pub, Session_key)
int Power_State;
string Encrypted_Neighbor_List; // E (Session_key, NBR_LST)
string Packet_MAC_Src_CH; //Mac (SH_ACH, P_SEQ_NUM | ID | E (CHK_pub, Session_key)
| Power_State | Encrypted_Neighbor_List)
  };
```

Data Structure 2: NBR_LIST packet data type.

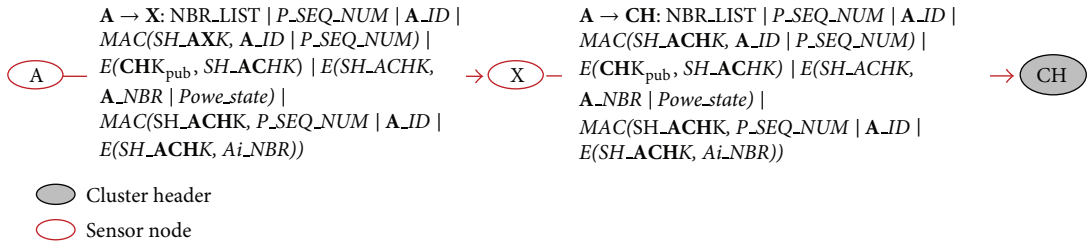


Figure 9: NBR_LIST packet.

$i \rightarrow j$: *NBR_LIST* | *P_SEQ_NUM* | *i_ID* | *MAC* (*SH_ijK, i_ID* | *P_SEQ_NUM*) | *E* (CHK$_{pub}$, *SH_iCHK*) | *E* (*SH_iCHK, i_NBR* | *Powe_state*) | *MAC* (SH_*ij*K, *P_SEQ_NUM* | *i_ID* | *E* (*SH_iCHK, i_NBR*)).

Each node has its neighbor list as a result from applying the L-PKI protocol. Each node receiving NBR_LIST packet does the following (*i* is the previous node, and *j* is the current node):

(1) check if node has received this NBR_LIST by searching *P_SEQ_NUM* in the table received_pkts; if the packet has been received once, then drop this packet; Otherwise store *P_SEQ_NUM* in received_pkts table of the node;

(2) compute the MAC value of *i*_ID and *P_SEQ_NUM* by using the shared key SH_*ij*K MAC (SH_*ij*K, *i*_ID | *P_SEQ_NUM*) and comparing it with the MAC value of the NBR_LIST; if equal, it moves to the next step; otherwise drop the packet;

(3) compute the MAC value of *j_ID* and *P_SEQ_NUM* by using the shared key between the *j* node and the node, which sends the *NBR_SEN* packet;

(4) send the *NBR_LIST* packet to node, which sends the *NBR_SEN* packet, until the packet reach the CH;

(5) if the current node is the CH, then decrypt "*E* (CHK$_{pub}$, *SH_iCHK*)" by using its private key, check the integrity by computing the MAC of *MAC* (*SH_iCHK*, *P_SEQ_NUM* | *i_ID* | *E* (*SH_iCHK, i_NBR*)), and compare it with the MAC value of the NBR_LIST packet. If the MAC values are equal, then CH decrypts neighbor information using the session key between the sender node and the CH; otherwise drop the packet. The format "data type" of the NBR_LIST packet is illustrated in Data Structure 2. The following packets and Figure 9 explain this operation:

A $\rightarrow$ ***X*** $\rightarrow$ ***CH***,

A $\rightarrow$ ***X***: NBR_LIST | *P_SEQ_NUM* | ***A***_*ID* | *MAC* (*SH_**AX**K, **A**_ID* | *P_SEQ_NUM*) | *E* (**CH**K$_{pub}$, *SH_**AC**HK*) | *E* (*SH_ACHK*, ***A***_*NBR* | *Powe_state*) | *MAC* (SH_**ACH**K, *P_SEQ_NUM* | ***A***_*ID* | *E* (*SH_**ACH**K, Ai_NBR*)),

A $\rightarrow$ ***CH***: NBR_LIST | *P_SEQ_NUM* | ***A***_*ID* | *MAC* (*SH_**ACH**K, **A**_ID* | *P_SEQ_NUM*) | *E* (**CH**K$_{pub}$, *SH_**AC**HK*) | *E* (*SH_ACHK*, ***A***_*NBR* | *Powe_state*) | *MAC* (SH_**ACH**K, *P_SEQ_NUM* | ***A***_*ID* | *E* (*SH_**ACH**K, Ai_NBR*)).

6.1.3. NBR_LIST Receiving. As we have said, if the current node is the CH, then decrypt "*E* (CHK$_{pub}$, *SH_iCHK*)" by using its private key, check the integrity by computing the MAC of *MAC* (*SH_iCHK, P_SEQ_NUM* | *i_ID* | *E* (*SH_iCHK, i_NBR*)), and compare it with the MAC value of the NBR_LIST

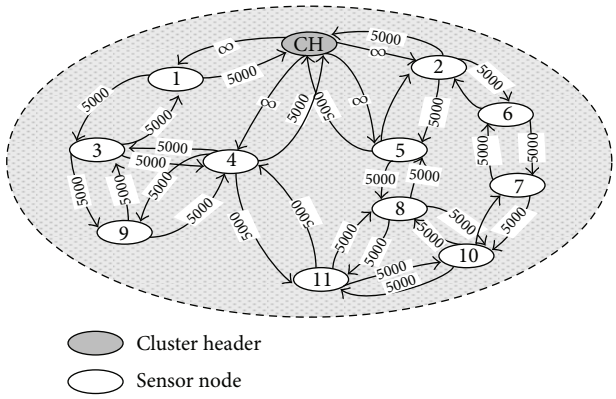


Figure 10: The initial weighted directed graph of one cluster.

packet. If the AMC values are equal, then CH decrypts neighbor information using the session key between the sender node and the CH; otherwise, drop the packet. After each CH has received neighbors' information and the state of power of all nodes on the cluster, each CH has a "vision" of the topology of the cluster nodes. Thus, CH is able to find secure disjoint multipath or partially disjoint multipath between each node and CH and select one of them according to specific algorithm.

6.1.4. Finding Secure Disjoint/Partially Disjoint Multipath. According to neighbors information each CH constructs a weighted directed graph and finds the multipath from the CH to every source node. SECMRP finds disjoint/partially disjoint multipath and selects N of them according to the minimum hops or the maximum energy available for each node on the path. Figure 10 shows a weighted directed graph G (V, E) of one cluster in the proposed WSNs. The weight of an edge in the corresponding graph of the network represents the available energy on the source (head) node. Due to the high energy of the CH, its weight is infinite. We assume that all other edges have the same weights 5000 mJ (the power state, which was sent from each node) after the CH has received NBR_LIST packets of all nodes.

As mentioned before, when the CHs receive the NBR_LIST packets, they start to calculate the shortest path among the available multipath to each node. The shortest path has the minimum sum of energy consumed for transmission of the packet, which is the path with minimum hops or the path with maximum energy available for each node. SECMRP modifies the Breadth First Search (BFS) algorithm (MBFS), similar to [14], but SEEM selects one path while SECMRP selects two paths. The main points of MBFS algorithm are as follows (assuming we need to transmit data from node 4 to CH, as shown in Figure 11):

(1) whenever it finds "$N = 2$" shortest two paths to node number 4, the MBFS algorithm checks if the weight of each edge on the paths is greater than the predefined level (SECMRP uses five define levels of energy limitation, which is similar to SEEM [14]. Each level is twice of the lower level). If the weight of each edge on the path is greater than the predefined level MBFS returns these paths as the shortest paths. Otherwise go to next step;

(2) if there is any node in the path, whose weight is less than the predefined level, MBFS does not select it and continues the searching until it finds the second shortest path. Thus, MBFS tries to balance the energy cost in the whole cluster;

(3) if MBFS cannot find any shortest path under current level of the energy limitation, it means that each path has at least one node whose energy is less than current energy limitation;

(4) thus MBFS algorithm uses the lower energy limitation and resumes the searching operation with the new level of energy limitation;

(5) if MBFS algorithm cannot find any path under any level of the energy limitation, it means that node 4 is unreachable and the CH cannot get the data from this node.

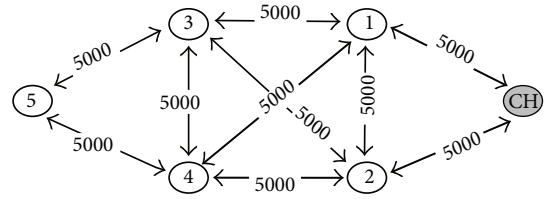


Figure 11: Weighted directed graph of on cluster in the WSN.

Periodically, each node sends its power state to the CH, and CH performs the MFBS for all nodes in the cluster and selects the best two paths between each node and CH then sends them to the corresponding node.

In the previous example, the energy limitation levels are 2500, 1250, 652, 312, and 156. We can realize that the path from source node to the CH goes through different nodes. Therefore, sometimes under specific energy limitation levels, we cannot find a path to specific node, but we can find multipath for other sensor nodes. Thus, each node must have its own energy limitation array. The maintaining and updating operations of this energy limitation array must be independent to each node. This ensures a high lifetime for the sensor network by balancing the energy cost.

6.2. Secure Data Transmission. In the secure route discovery phase each CH is responsible for finding secure disjoint multipath or partially disjoint multipath between each node and its CH. Thus, each node becomes able to transmit its sensed data securely. After the secure route discovery phase and upon the application. CH must have sent the best ($P = 2$) energy efficient paths to each node. Now, nodes become able to send the sensed data to the CH. Secure data transmission phase has two steps:

(1) in the first step, the sensor node encrypts and sends data to the CH;

(2) the second step takes place at the CH; CH aggregates, compresses the received data, which can be from more than one node, and sends the aggregated data to the BS.

By applying the MBFS algorithm, the paths will be "$4 \rightarrow 2 \rightarrow CH$" and "$4 \rightarrow 1 \rightarrow CH$". According to the paths, the source node divides the data to N parts "in this example $p = 2$", and makes the following two packets, and sends them to the nodes number 1 and 2. The data packets "DATA_PACKET" have the following formats:

$$\textbf{\textit{4} $\rightarrow$ \textit{1}}: \textit{DATA\_PACKET} \mid \textit{P\_SEQ\_NUM} \mid \textit{4\_ID} \mid \textit{Path} \mid \textit{MAC}(\textit{SH\_41K}, \textit{P\_SEQ\_NUM} \mid \textit{4\_ID} \mid \textit{Path}) \mid E(\textit{SH\_4CHK}, \textit{DATA\_PART1}) \mid \textit{MAC}(\textit{SH\_4CHK}, \textit{P\_SEQ\_NUM} \mid \textit{4\_ID} \mid E(\textit{SH\_4CHK}, \textit{DATA\_PART1})),$$

$$\textbf{\textit{4} $\rightarrow$ \textit{2}}: \textit{DATA\_PACKET} \mid \textit{P\_SEQ\_NUM} \mid \textit{4\_ID} \mid \textit{Path} \mid \textit{MAC}(\textit{SH\_41K}, \textit{P\_SEQ\_NUM} \mid \textit{4\_ID} \mid \textit{Path}) \mid E(\textit{SH\_4CHK}, \textit{DATA\_PART2}) \mid \textit{MAC}(\textit{SH\_4CHK}, \mid \textit{P\_SEQ\_NUM} \mid \textit{4\_ID} \mid E(\textit{SH\_4CHK}, \textit{DATA\_PART2})).$$

To avoid several type of attacks and to balance the power conception in the network, SECMRP divides the data to two parts and sends them via two disjointed paths "$4 \rightarrow 2 \rightarrow CH$ and $4 \rightarrow 1 \rightarrow CH$."

As we can see in the first packet the 4_ID is the ID of the source node. SECMRP uses the *MAC* (*SH_* ***41K***, *P_SEQ_NUM* | *4_ID* | *Path*) to insure the *authentication* between it and node number 1 and to avoid several types of attacks. SECMRP ensures the *confidentiality* by encrypting the *DATA_PART1* by using the shared key of node 1 and CH *E* (*SH_ACHK, DATA_PART1*). To insure the integrity between the source node 1 and the destination "CH," SECMRP uses the MAC function and using the shared key of source node and the CH (*MAC* (*SH_ACHK,* | *P_SEQ_NUM* | ***4*** *_ID* | *E* (*SH_ACHK, DATA_PART1*))).

The following section explains the operation of one packet, and the other packet has the same operation. Each intermediate node, which receives the *DATA_PACKET* packet, does the following things:

(1) checks if it has received this *DATA_PACKET* by searching *P_SEQ_NUM* in the received_pkts table; if the packet has been received once, then drop this packet. Otherwise store *P_SEQ_NUM* in received_pkts table and move to next step;

(2) computes the MAC value of *4_ID, P_SEQ_NUM,* and *Path* by using the shared key SH_41K MAC (SH_41K, *4_ID* | *P_SEQ_NUM* | *Path*) and compares it with the MAC value of the DATA_PACKET packet. If they are equal, it moves to next step;

(3) computes the MAC value *4_ID, P_SEQ_NUM.* In the first step, the source sensor node sends the data packet to the CH via the secure path that CH has sent to the source node.

Let us take the example of Figure 11. The initial multipath from the source node 4 to CH is illustrated in Table 4. Assume that the number of paths, which is needed to send data is two $P = 2$ and the levels of energy limitation are 2500, 1250, 652, 312, and 156. And *Path* MAC (SH_1CHK, *4_ID* | *P_SEQ_NUM* | *Path*) by using the shared key between it and next node in the path "*i*", in this example CH "SH_1CHK". And add it to the *DATA_PACKET* packet. The following packet explains this step:

$$\textit{1} \rightarrow \textit{CH}: \textit{DATA\_PACKET} \mid \textit{P\_SEQ\_NUM} \mid \textbf{\textit{4}}\textit{\_ID} \mid \textit{MAC}(\textit{SH\_}\textbf{\textit{1CHK}}, \textit{P\_SEQ\_NUM} \mid \textit{4\_ID} \mid \textit{Path}) \mid E(\textit{SH\_4CHK}, \textit{DATA\_PART1}) \mid \textit{MAC}(\textit{SH\_}\textbf{\textit{4CHK}}, \textit{P\_SEQ\_NUM} \mid \textbf{\textit{4}}\textit{\_ID} \mid E(\textit{SH\_4CHK}, \textbf{\textit{DATA\_PART1}}));$$

(4) sends the new *DATA_PACKET* packet to the next node "*next hop*" in the path;

(5) at the CH, CH performs step 2, 3 and checks the integrity by comparing the MAC values *MAC* (*SH_****4CHK****, P_SEQ_NUM* | ***4*** *_ID* | *E* (*SH_4CHK,* ***DATA_PART1***)), and decrypts *E* (*SH_****4****CHK,* ***DATA_PART1***) using the shared key between source node and CH *D* (*SH_****4****CHK, E* (*SH_****4****CHK,* ***DATA_PART1/2***)).

After the decryption operation, CH sends an acknowledgement (ACK) to the source node. The format "data type" of the DATA_PACKET packet is illustrated in Data Structure 3.

The second step starts after the data becomes available at the CH. CH becomes able to aggregate and compress the received data, which can consist of more than one node. After aggregation and compression operations, the CH encrypts the aggregated data and sends it to BS. The following packet illustrates the aggregated data packet:

$$\textbf{\textit{CH}} \rightarrow \textbf{\textit{BS}}: \textit{DATA\_PACKET} \mid \textit{P\_SEQ\_NUM} \mid \textbf{\textit{CH}}\textit{\_ID} \mid \textit{MAC}$$

Table 4: All available paths from node 4 to CH.

Seq.	Path	Available energy	Energy limitation
1	$4 \rightarrow 1 \rightarrow$ CH	5000-5000	2500
2	$4 \rightarrow 2 \rightarrow$ CH	5000-5000	2500
3	$4 \rightarrow 2 \rightarrow 1 \rightarrow$ CH	5000-5000-5000	2500
4	$4 \rightarrow 1 \rightarrow 2 \rightarrow$ CH	5000-5000-5000	2500
5	$4 \rightarrow 3 \rightarrow 2 \rightarrow$ CH	5000-5000-5000	2500
6	$4 \rightarrow 3 \rightarrow 1 \rightarrow$ CH	5000-5000-5000	2500

```
packet Secure_Date_Packet
  {
string Packet_Type;
string P_SEQ_NUM;
int Node_ID;
string Path;
string Packet_MAC_Neighbors;
string Encrypted_Data;
string Packet_MAC_Src_CH;
  };
```

Data Structure 3: DATA_PACKET data type.

(*SH_**CHBSK**, P_SEQ_NUM* | ***CH**_ID* | *Path*) | *E* (*SH_**CHBS**K, DATA*) | *MAC* (*SH_**CHBS**K, P_SEQ_NUM* | ***CH**_ID* | *E* (*SH_**CHBS**K, DATA*)).

From the *DATA_PACKET* packet, we can realize that the communication is between the CH and the BS. In this step, SECMRP ensures the authentication and integrity between the CH and BS using MAC and ensure confidentiality by encrypting the *DATA_PACKETE (SH_**CHBS**K, DATA)* using the shared key between CH and the BS.

6.3. Route Maintenance. In SECMRP, each CH has a vision of the topology of its members, and knows the available energy of each node. All this information allows the CH to become able to maintain the route as needed. As discussed, each CH constructs a weighted directed graph and finds the multipath from the CH to every source node by using the MBFS algorithm. The initial information of the power state of each node has been received with the NBR_LIST packet. In the route maintenance phase, each node sends periodically the state of its power to the CH, so each CH can get updated information of power state of each node on the cluster.

SECMRP can maintenance the route by decreasing the weight when the source node sends or receives packets. If the available energy of the node reaches to the energy limitation level, then it uses another path. It means that if one node on the shortest path has energy less than the specific level, then the MBFS discards this path and continues searching for the second shortest path.

7. Experimentation

To validate our proposed protocols, we used OMNeT++4.1 as a simulator, C++ as a programming language, and Crypto++ 5.6.1 as a library for cryptographic algorithms. OMNeT++ provides an integrated development environment based on the eclipse platform and C++ simulation library and framework [20]. OMNeT++ is used for simulating wired and wireless communication networks. Crypto++ 5.6.1 is a C++ library for cryptographic algorithms which implements ECC, AES-128, ECDSA-160 [16, 18, 21], and the other crypto functions needed by the protocols.

7.1. L-PKI Experimentation Results. L-PKI is tested in two different scenarios: the first one was with 10 sensor nodes, two CHs and one BS, and the second scenario was with 5 sensor nodes, one CHs and one BS. In both scenarios, nodes are connected with a channel that has a data rate of 250 Kbps and 100 Mbps for CHs and base stations. Figure 12 illustrates the experimental work of L-PKI protocol.

Energy Cost in L-PKI. We can estimate the energy cost of any key management scheme by calculating the energy required for the execution of cryptographic operation and the energy needed for transmitting the required data. There are many studies, which concentrate on computing the energy cost in WSNs [4, 6, 16, 19]. Each study uses different models. To analyze the energy cost of L-PKI, we can compute the energy cost of any model such as MICAz [22], MICA2DOT [23], and TelosB [30]. MICAz mote is a third-generation module based on the low power 8 bit ATmega128L microcontroller with a clock frequency of 7.37 MHz [19, 22]. It is used to enable low power, wireless sensor network. MICAz runs TinyOS and embed 2.4 GHz IEEE 802.15.4 radio with a claimed data rate of 250 kbps. MICAz offers hardware security (AES-128) [22]. TelosB is a mote module based on the low power 16 bit MSP430 microcontroller with 10 kB RAM and a clock frequency of 4 MHz [19, 24]. TelosB runs TinyOS and embed IEEE 802.15.4 radio with data rate of 250 kbps. Table 5 shows the power which MICAz, TelosB, and MICA2DOT needs for computation and transmission operations [16, 19]. Table 6 illustrates the cost of each computation operation. In Table 6, MICAz running at 7.37 MHz and TelosB at 4 MHz for application data rates, respectively 108 kbps and 75 kbps [19]. Also Table 5 shows the equivalence number of cycles of computation, which is indicated in parenthesis for each operation.

We are going to use the MICAz and TelosB [19] models to calculate the required power. According to [19], the transmission of a single bit of data requires 0.60 μJ and 0.67 μJ for reception. We can calculate the energy cost of *L-PKI* by calculating the energy required for the execution of cryptographic operation and the energy needed for transmitting the required data. We can conclude the energy consumption of L-PKI for each node in Table 7.

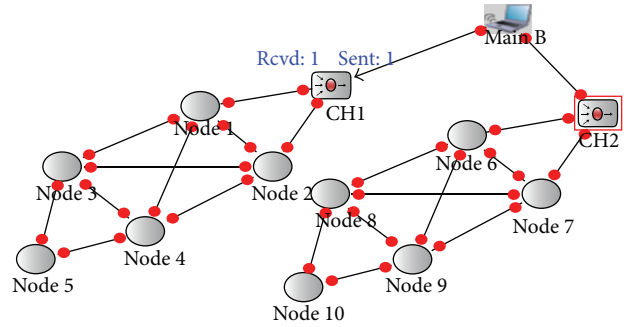


Figure 12: First step in the authentication of L-PKI (first scenario).

Table 5: Energy costs of MICAz, TelosB, and MICA2DOT [16, 19].

Energy cost	MICAz	TelosB	MICA2DOT
Compute for $1T_{clk}$	3.5 nJ (1)	1.2 nJ (1)	
Transmit 1 bit	0.60 μJ (170)	0.72 μJ (600)	59.2 μJ/byte = 7.4 μJ/bit.
Receive 1 bit	0.67 μJ (190)	0.81 μJ (680)	28.6 μJ/byte = 3.5756 μJ/bits.

Table 6: Cost of computation operation.

Energy cost	MICAz	TelosB
AES-128 128-bit encrypt	38 μJ	9 μJ
ECC-160	55 mJ	17 mJ
ECDSA-160 sign	52 mJ	15 mJ
ECDSA-160 verify	63 mJ	19 mJ

Table 7: Total energy cost of L-PKI for each node.

Node	Operation	Required energy	
		MICAz	TelosB
Transmission	Send digital certificate	$192 * 0.6\,\mu J = 115.2\,\mu J$	$192 * 0.72\,\mu J = 138.2\,\mu J$
	Receive neighbor's digital certificate	$192 * 0.67 = 128.6\,\mu J$	$192 * 0.81\,\mu J = 155.5\,\mu J$
	Send signature	$320 * 0.6\,\mu J = 192\,\mu J$	$320 * 0.72 = 230.4\,\mu J$
	Receive neighbor's signature	$320 * 0.67 = 214.4\,\mu J$	$320 * 0.81\,\mu J = 259.2\,\mu J$
	Send or receive session key	$128 * 0.67\,\mu J = 85.76\,\mu J$	$128 * 0.81\,\mu J = 103.68\,\mu J$
	Send OK message	$16 * 0.6\,\mu J = 9.6\,\mu J$	$16 * 0.72\,\mu J = 11.52\,\mu J$
	Receive OK message	$16 * 0.67\,\mu J = 10.72\,\mu J$	$16 * 0.81\,\mu J = 12.96\,\mu J$
	Total cost	**756.28** μJ	**911.46** μJ
Computation	Verification operation of certificate	63 mJ	19 mJ
	Encryption/decryption session key	55 mJ	17 mJ
	Total cost	**118** mJ	**36** mJ
Total		**118.756** mJ	**36.911** mJ

Table 8: Total energy cost of SECMRP for each node.

Energy cost	MICAz	TelosB
Secure route discovery	56.667 mJ	18.934 mJ
Secure data transmission	845.54 μJ	908.28 μJ

7.2. SECMRP Experimentation Results. SECMRP is tested in two different scenarios. The first one was with 10 sensor nodes, two CHs and one BS. The second scenario was with 5 sensor nodes, one CHs and one BS. In both scenarios, nodes are connected with a channel that has a data rate of 250 Kbps and 100 Mbps for CHs and base stations. In the rest of this section, we are going to concentrate on the first scenario.

Energy Cost in SECMRP. We calculated the energy cost of SECMRP by calculating the energy required for the execution of cryptographic operations and the energy needed for transmitting the required data.

SECMRP has three phases secure route discovery, secure data transmission, and route maintenance. We are going to calculate the energy cost of each phase in SECMRP. We can conclude the energy consumption of SECMRP for each node in Table 8.

As we can see from Table 8, secure data transmission phase requires just 845.54 μJ in MICAz and 908.28 μJ in TelosB.

8. Discussion and Analysis

8.1. Protocol Security. The proposed protocols provide several security services, including mutual node authentications, confidentiality, and integrity for sensitive data, while balancing energy cost and performance. The proposed protocols prevent several attacks, such as passive attacks by dividing the sensed data into two parts and encrypting these parts. The protocols utilize multipath to make it difficult for the attacker to capture the whole data. The protocols protect against impersonation or spoofing attacks by invoking mutual authentication using certificates. Modification of protocol messages attacks can be avoided by authentication using nodes' certificates and integrity using MAC. The proposed method clusters the network and uses the certifications to avoided routing table overflow attacks. Replay attacks can be avoided in the proposed method using timestamp and checking the integrity.

Table 9: L-PKI, TinyPK, and μPKI comparison based on security services.

	Authentication	Confidentiality	Nonrepudiation	Scalability
L-PKI	√	√	√	√
TinyPK	√	√		
μPKI		√		√

8.2. Comparison with Related Works. Our proposed L-PKI protocol provides enhancments over other related work, like TinyPK and μPKI. In μPKI, the BS plays the main role in creating the session keys between each pair of nodes. As a result, a lot of traffic is created. In addition, a somewhat weak authentication exists between nodes in the network (as discussed in the Section 2). Table 9 provides a comparison between L-PKI, TinyPK, and μPKI, based on security services.

Our proposed SECMRP protocol enhances several security issues over other route discovery protocols like SEEM and INSENS. SECMRP uses the concept of multipath and clustering to deal with security and efficiency. SECMRP enhances the security issue of SEEM and at the same time uses two disjointed path to send sensed data to the CH. In SEEM, BS works as a server, which floods the query to the network and the node which satisfies the query will send a request to BS for a path, but CH in SECMRP periodically sends two paths to each node. Table 10 provides a comparison between SECMRP, SEEM, and INSENS based on security services.

9. Conclusion and Future Work

Our work improves existing related work by integrating a lightweight PKI with a secure data transmission service via multipath in an efficient manner in terms of resource and power consumption. The proposed protocols enhance the security of WSN by providing mutual authentication between

Table 10: SECMRP, SEEM, and INSENS comparison based on security services.

	Authentication	Confidentiality	Integrity	Balancing	Scalability
SECMRP	√	√	√	√	√
SEEM				√	
INSENS	√	√	√	√	

neighbor nodes and sending the data via multipath. We have provided extensive simulation and evaluation for the proposed protocols and contrasted them with related work. The results are encouraging.

In future work, we plan to enhance the MBFS algorithm to optimize power efficiency as much as possible. We also plan to actually deploy the proposed algorithms on actual sensors in a typical WSN environment.

Acknowledgment

This work is funded by the National Plan for Science and Technology at King Saud University, Project no. 11-INF1500-02.

References

[1] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective*, IEEE Press-Wiley, Hoboken, NJ, USA, 2009.

[2] J. Zhang and V. Varadharajan, "Wireless sensor network key management survey and taxonomy," *Journal of Network and Computer Applications*, vol. 33, no. 2, pp. 63–75, 2010.

[3] J. Sen, "A survey on wireless sensor network security," http://arxiv.org/abs/1011.1529.

[4] F. Amin, A. Jahangir, and H. Rasifard, "Analysis of public-key cryptography for wireless sensor networks security," *Proceedings of World Academy of Science, Engineering and Technology*, vol. 31, 2008.

[5] G. Gaubatz, J. P. Kaps, and B. Sunar, "Public key cryptography in sensor networks-revisited," in *Security in Ad-Hoc and Sensor Networks*, pp. 2–18, 2005.

[6] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, "Comparing elliptic curve cryptography and RSA on 8-bit CPUs," in *Cryptographic Hardware and Embedded Systems-CHES 2004*, pp. 925–943, 2004.

[7] R. A. Mollin, *An Introduction to Cryptography*, Chapman & Hall, Boca Raton, Fla, USA, 2006.

[8] M. Faisal, J. Al-Muhtadi, and A. Al-Dhelaan, "Towards efficient security services in wireless sensor networks," in *Computer Applications for Bio-technology, Multimedia, and Ubiquitous City Communications in Computer and Information Science*, vol. 353, pp. 114–123, 2012.

[9] B. Kadri, M. Feham, and A. M'hamed, "Lightweight PKI for WSNs uPKI," *International Journal of Network Security*, vol. 10, no. 3, pp. 194–200, 2010.

[10] V. S. Miller, "Use of elliptic curves in cryptography," in *Proceedings of the Advances in Cryptology (CRYPTO '85)*, H. C. Williams, Ed., vol. 218, pp. 417–426, Springer, Berlin, Germany, 1986.

[11] R. Watro, D. Kong, S. F. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, pp. 59–64, October 2004.

[12] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[13] W. Diffie and M. E. Hellman, "New directions in cryptography," in *Secure Communications and Asymmetric Cryptosystems*, pp. 143–180, Westview, Boulder, Colo, USA, 1982.

[14] N. Nasser and Y. Chen, "SEEM: secure and energy-efficient multipath routing protocol for wireless sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2401–2412, 2007.

[15] J. Deng, R. Han, and S. Mishra, "INSENS: intrusion-tolerant routing for wireless sensor networks," *Computer Communications*, vol. 29, no. 2, pp. 216–230, 2006.

[16] A. S. Wandert, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom '05)*, pp. 324–328, March 2005.

[17] E. Noroozi, J. Kadivar, and S. H. Shafiee, "Energy analysis for wireless sensor networks," in *Proceedings of the 2nd International Conference on Mechanical and Electronics Engineering (ICMEE '10)*, vol. 2, pp. 382–386, August 2010.

[18] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer, New York, NY, USA, 2004.

[19] G. De Meulenaer, F. Gosset, F. X. Standaert, and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks," in *Proceedings of the 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communication (WiMob '08)*, pp. 580–585, October 2008.

[20] OMNeT++ Community, http://www.omnetpp.org/.

[21] The AES-CMAC Algorithm, http://tools.ietf.org/html/rfc4493.

[22] MICAz, http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html.

[23] MICA2DOT, http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html.

[24] TELOSB, http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 406061, 7 pages
http://dx.doi.org/10.1155/2013/406061

Research Article

A Secret Sharing-Based Key Management in Hierarchical Wireless Sensor Network

Yiying Zhang,[1] Chunying Wu,[2] Jinping Cao,[1] and Xiangzhen Li[3]

[1] *State Grid Information & Telecommunication Company Ltd., Beijing 100761, China*
[2] *Computer Science School of Shaanxi Normal University, Xi'an, Shaanxi 710062, China*
[3] *State Grid Electric Power Research Institute, Nanjing, Jiangsu 210003, China*

Correspondence should be addressed to Yiying Zhang; zhangyiying1973@hotmail.com

Received 31 January 2013; Revised 13 May 2013; Accepted 13 May 2013

Academic Editor: Muhammad Khurram Khan

Wireless sensor networks (WSNs) are subject to various attacks because of the vulnerable environment, limited recourse, and open communication channel. To protect WSNs, in this paper, we present a Secret sharing-based key management (SSKM). SSKM utilizes the advantages of hierarchical architecture and adopts two-level key management and authentication mechanism, which can efficiently protect the allover network communication security and survivability. Different from previous works, the SSKM distributes keys based on secret sharing mechanism by the clustered architecture, which not only localizes the key things but also keeps scalability. The SSKM provides various session keys, the network key for base station (BS) and cluster heads (CHs); the cluster key between the cluster head and member nodes. The SSKM dynamically generates different keys based on different polynomials from BS in different periods which can protect the network from the compromised nodes and reduce the high probability of the common keys. The security analysis shows that the SSKM can prevent several attacks effectively and reduce the energy consumption.

1. Introduction

Due to the development of internet of things (IoT) and cyber physical system (CPS), wireless sensors have been deployed in many applications, such as in smart grid, national security, intelligent transportation, forest detection, or chemical harmful gas monitoring [1]. However, wireless sensor networks (WSNs) usually consist of tiny sensors which have low computational capability, small storage, and limited energy; that is, the WSNs are often subject to a variety of attacks, such as eavesdropping attack and flood attack and so on. Once a sensor is compromised by adversaries, the information materials of the sensor become non secretive and intercepted by enemy, and the entire network is threatened [2].

Therefore, security mechanisms in WSN are required to provide data confidentiality, integrity, freshness, availability, and authentication [2]. Moreover, in view of the excellent performance of the clustering algorithm in WSN, the hierarchical architectures are often used in WSN applications [3]. Normally, cryptographic methods of securing a network are the key management strategy, and it has been intensively studied in the literature of WSNs [4–13]. Therefore, some literatures adopt hierarchical architecture to deploy key system on them to protect the communication in WSNs [4–10].

In [5, 6], the authors employed the secret sharing mechanism to distribute keys into nodes, which can effectively generate and assign keys. However, in these schemes, the network must exchange many messages to establish key system, which consumes lots of energy. In this paper, we present a novel secret sharing-based key management (SSKM).

In SSKM, considering that the energy efficiency is a dominant consideration problem of WSNs, we firstly employ the maximum energy cluster head (MECH) protocol to form cluster. Different from other hierarchical architectures, MECH protocol limits the size of cluster to generate uniform cluster. In each cluster, there is a sensor, called cluster head (CH), collecting information from other cluster member nodes and forwarding the processed information to the base station.

Therefore, to protect the communication channel from CH to BS, we present a network key. Firstly, the BS encrypts

the network key with a secret, puts the secret as constant of the polynomial, and divides the secret into shares based on Lagrange interpolation formula. To reconstruct the polynomial of $(t-1)$ degree, any t or more parameters (ID, $f(\mathrm{ID})$) combination can recover and obtain the secret. Therefore, our solution tries to avoid the adversary intercept and capture sufficient parameters.

Also, similar to network key, we design a cluster key to protect the communication between the cluster head and member nodes. Unlike the BS, CHs have no sufficient energy to broadcast messages. Thus, the BS deploys key material to sensors in advance, such as polynomials and revoked list. Then, the CH just exchanges parameters to adjust polynomials to generate/cancel keys.

Compared to previous works [5, 6], the salient advantages of our work are as follows:

(1) SSKM establishes a relocatable key mechanism based on the secret sharing theory, which hides keys into secret and recovers them when needed;

(2) SSKM adopts hierarchical architecture which is suitable for the secret sharing mechanism and localizes the security and reduces energy consumption. It makes the SSKM key management feasible;

(3) SSKM presents an authentication mechanism based on the secret sharing theory, which supports the scalability (join or leave).

The rest of this paper is organized as follows. Section 2 describes the related work, Section 3 presents the system model and assumption, Section 4 describes the secret sharing key management in detail, and Section 5 evaluates SSHM using security analysis. Finally, we end the paper with a conclusion as well as the further work in Section 6.

2. Related Work

In 1979, Shamir [14] and Blakley [15] proposed the secret sharing method based on the Lagrange interpolation formula and the nature of the vector space, respectively. Proposition: given $n+1$ points $(x_i, f(x_i))$ on a polynomial $f(x)$ of degree n, one can identify a uniquely polynomial by calculating:

$$f(x) = \sum_{i=1}^{n} \left(f(x_i) \coprod_{1 \le k \ne i \le n} \frac{x - x_k}{x_i - x_k} \right). \quad (1)$$

One also defines the Lagrange coefficient $\Delta_{i,s}$ for $i \in Z_P$ and a set $S \subseteq Z_P$:

$$\Delta_{i,s}(x) = \coprod_{i \in S, j \ne i} \frac{x - j}{i - j}. \quad (2)$$

A (t, n) threshold secret sharing scheme is as follows: given n points $(x_i, f(x_i))$ on a polynomial $f(x)$ of degree $(t-1)$, randomly picking out t points of $(x_i, f(x_i))$ from n points can construct the polynomial. When constructing the scheme, the credibility of the parties splits initial secret S into n shares (points) and assigns them to users safely. Any t or more users combining their share can reconstruct the secret S, but any $t-1$ user group or less cannot reconstruct the secret. This secret sharing method provides the security scheme in many applications, such as key distribution, secure computation, and information safe storage.

In [5], the authors present a low-cost secret-sharing scheme for sensor network. This paper provides basic building blocks to establish secure communication through exchanging secret keys between neighbor nodes without any cryptography methods. In [5], authors also design a second algorithm which extends the secret key establishment. However, due to the exchange happening among sensors, it consumes lots of energy. Moreover, the authentication between neighbor nodes also needs to exchange large messages, which makes it unsuitable for wireless sensor network.

In [6], authors presented some schemes to secure data aggregation based on secret sharing and information dispersal. In these schemes, sensor nodes split messages into subshares and forward them among several disjoint paths to defend DoS attack, eavesdropping attack, and tampering attack. They design a secret multipath aggregation (SMA) mechanism which applies secret sharing to create shares to deal with security under the contingency of node compromise. However, these schemes are not feasible for heavy energy consumption. On one hand, they want data aggregation using secret sharing; on the other hand, they have to distribute key things and messages to confuse the enemy, so that the adversary cannot find the real route, which needs a large number of messages exchange.

Comparing with previous works, our solution adopts the hierarchical network and localizes the communication and security. Also, we ingeniously use the base station to carry complicated things out, which can reduce the energy consumption.

3. System Model

3.1. Network Model. The wireless sensor network is energy sensitive. Therefore, we adopt the maximum energy cluster head (MECH) protocol for our network architecture [1]. The MECH is an LEACH-like protocol (LEACH: low energy adaptive clustering hierarchy) [2] which divides the network into clusters.

As shown in Figure 1, in the MECH architecture, the sensors self-organize into some clusters and act as two types of roles: cluster heads and member nodes. In each cluster, one node as a CH manages the cluster and deals with information from member nodes forward to the base station (BS). MECH constructs clusters based on radio range and the number of cluster members. The cluster topology in the network is distributed more equally through our cluster constructing; that is, nodes in each cluster do not exceed a certain threshold.

3.2. Assumptions. In the considered network, we consumed the following.

(i) All sensor nodes are static.

(ii) Each sensor has a unique ID assigned by the base station.

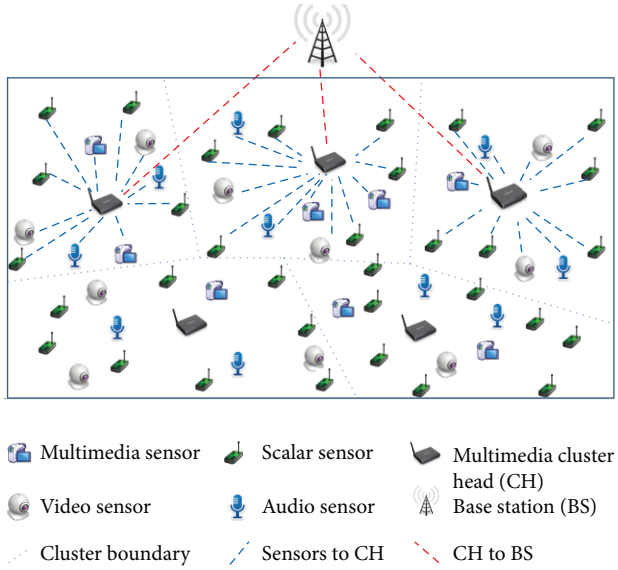


Figure 1: The network system.

Table 1: Notations.

Notation	Explains
K_{ini}	The initial key shared by all nodes
ID_i	ID of sensor
BS	Base Station
C_i	The ith cluster
CH_i	The ith cluster head
$K^l_{C_{\text{in}}}$	The session key during session l
$S^l_{C_{\text{in}}}$	The secret sharing among CHs during session l
$Z^l_{C_{\text{in}}}$	The encrypted key $K^l_{C_{\text{in}}}$ by the secret sharing $S^l_{C_{\text{in}}}$
$K^l_{\text{CH}_i}$	The cluster key among clusters during session l
$S^l_{\text{CH}_i}$	The secret sharing between CH and members in session l
$Z^l_{\text{CH}_i}$	The encrypted key $K^l_{\text{CH}_i}$ by the secret sharing $S^l_{\text{CH}_i}$
R	Revoked set before session l
l	Session period
w	The number of nodes in WSN

(iii) Each sensor has the same capabilities in energy, computation, radio range, and so forth.

(iv) If a node is compromised, all of the key things in the node are revealed [7].

(v) Each sensor is in, and only in, one cluster.

(vi) The BS can communicate with all sensors in the network.

3.3. Notations. In Table 1, we list some notations used in this paper.

4. The Secret Sharing-Based Key Management

In this section, we describe the secret sharing-based key management (SSKM) in detail. After deployment, the base station assigns each sensor an initial key K_{init} similar to LEAP+ [12]. And then, the BS broadcasts the key materials to the network to build the network key and the cluster key, respectively. The key architecture is shown in Figure 2.

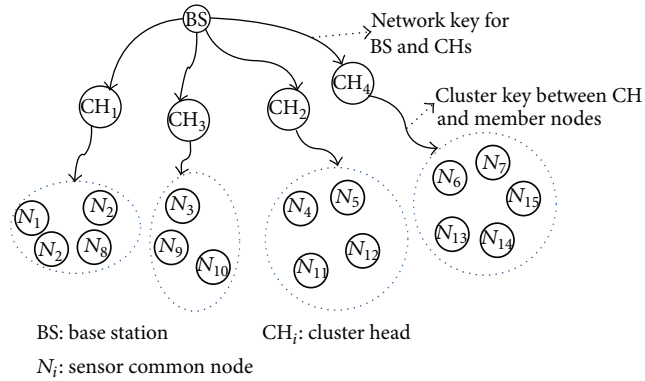


Figure 2: The key architecture.

4.1. Preliminaries. After deployment, the BS randomly chooses an integer, relatively primes with $p-1$ and $q-1$ (p and q are big primes); the base station assigns each sensor an initial key K_{init} similar to LEAP. Since the BS is credible and has sufficient capacity and energy, it keeps all IDs and keys. Meanwhile, BS chooses polynomials and the corresponding number of polynomial values for each cluster. Assume that there are $m-1$ clusters C_i $(i = 1, \ldots, m-1)$, and each cluster has a cluster head CH_i and k $(k \geq t)$ member nodes.

Shamir's (t, n) threshold scheme based on Lagrange interpolating polynomial states that there are n shareholders $U = \{U1, \ldots, U_n\}$ and mutually trusted dealer D. The scheme consists of two steps [16]:

(1) *share generation phase*: dealer D randomly selects a polynomial $f(x)$ of $(t-1)$ degree:

$$f(x) = S + a_1 x + \cdots + a_{t-1} x^{t-1} \pmod{P}, \tag{3}$$

in which the secret $S = f(0)$, all coefficients $S, a_1, a_2, \ldots, a_{t-1}$ are in finite field $F_p = \text{GF}(P)$ with p elements, and dealer computes all shares $S_i = f(x_i)$ for $\{i = 1, \ldots, n\}$; then it distributes each share S_i to corresponding shareholder U_i privately;

(2) *secret reconstruct phase*: any t shares $(S_1, S_2, \ldots, S_t)$ of n shares as input, and we can reconstruct the secret S as

$$f(x) = \sum_{i=1}^{t} S_i \left(\prod_{i \neq j} \frac{x - x_j}{x_i - x_j} \right) \pmod{P},$$
$$f(0) = \sum_{i=1}^{t} S_i \left(\prod_{i \neq j} \frac{x_j}{x_j - x_i} \right) \pmod{P}. \tag{4}$$

We find that the above scheme satisfies the basic security requirement of secret sharing scheme: any t shares or more than t shares can reconstruct the secret S; fewer than $(t-1)$ shares cannot reconstruct the secret S. Shamir's scheme is information theoretically secured [17]. However, there are some requirements [18] in this situation:

(1) there must be a secure channel for delivering shares between dealer and users;

(2) x_i and $f(x_i)$ are made publicly known. However, in key transfer protocol, for security reason, we need to keep x_i and $f(x_i)$ as each user's secret. So we adopt the discrete logarithm in finite field and DDH difficulty assumption to ensure the security in the unsecure communication channel.

4.2. Initial Phase. Once wireless sensor network has been deployed and sensors self-organized into clusters, BS starts to form the key system as follows.

(1) Firstly, BS chooses two big primes p_1 and q_1; let $p = 2p_1 + 1$ and $q = 2q_1 + 1$, $N = pq$; it is computationally intractable to solve the factor n without p, q. Meanwhile, BS selects a generator g ($g \in [N^{1/2}, N]$) and another prime Q ($Q > N$). And then, BS broadcasts the three triple (N, g, Q) to sensors in the network.

(2) Assume that during each session period l ($l = 1, \ldots, M$), BS randomly and uniformly chooses m polynomials $f(x)$ of $(t-1)$-degree, where $m - 1 \geq t$. And one of polynomials is as follows:

$$f^l_{C_{\text{in}}}(x) = S^l_{C_{\text{in}}} + a^l_{1,C_{\text{in}}} x + \cdots + a^l_{t-1,C_{\text{in}}} x^{t-1} \bmod Q. \quad (5)$$

Equation (5) is used to key distribute between BS and cluster head. And other $m-1$ polynomials are utilized to key distribute among the cluster and member nodes as follows:

$$f^l_{\text{CH}_i}(x) = S^l_{\text{CH}_i} + a^l_{1,\text{CH}_i} x + \cdots + a^l_{t-1,\text{CH}_i} x^{t-1} \bmod Q. \quad (6)$$

(3) BS independently selects M session keys $\{K^l_{C_{\text{in}}}\}_{l=1,\ldots,M}$ and $\{K^l_{\text{CH}_i}\}_{l=1,\ldots,M}$ from GF(Q) in the finite field Q and hides the session keys with secret $S^l_{C_{\text{in}}}$ and $S^l_{\text{CH}_i}$, namely, $Z^l_{C_{\text{in}}} = \{K^l_{C_{\text{in}}} + S^l_{C_{\text{in}}}\}$ in network key management and $Z^l_{\text{CH}_i} = \{K^l_{\text{CH}_i} + S^l_{\text{CH}_i}\}$ in cluster key management. The algorithm of initial phase is shown in Figure 3.

4.3. Network Key Management. The network key is the session key between the BS and cluster heads to protect their communication.

The key shares distribution process is as follow.

(1) During the session period $\{l\}_{l=1,\ldots,M}$, the BS counts out each cluster head's (CH's) share $f^l_{C_{\text{in}}}(\text{ID}^l_{\text{CH}_i})$ by their ID and gets the two tuples $\{\text{ID}^l_{\text{CH}_i}, f^l_{C_{\text{in}}}(\text{ID}^l_i)\}$, for the security between BS and CHs. Firstly, the BS randomly chooses x^l_0 ($x^l_0 \in [2, n]$) which relatively primes with $p-1$ and $q-1$, and let $y^l_0 = g^{x^l_0} \bmod N$; meanwhile, CHs also chooses x^l_i ($x^l_i \in [2, N]$) and lets $y^l_i = g^{x^l_i} \bmod N$. And then, the CH sends $(\text{ID}^l_{\text{CH}_i}, y^l_i)$ ($i = 1, \ldots, m-1$) to the BS, and BS unicasts $(y^l_0, (\text{ID}^l_{\text{CH}_i}, f^l_{C_{\text{in}}}(D^l_{\text{CH}_i})(y^l_{\text{CH}_i})^{x_0} \bmod N$ to each $\text{ID}^l_{\text{CH}_i}$. Note that the BS must ensure if $\text{ID}_i \neq \text{ID}_j$ then $y^l_i = y^l_j$; otherwise, it should regenerate until success.

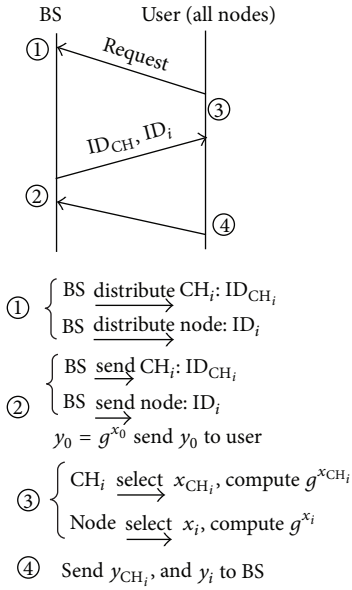


Figure 3: The initial phase.

(2) Given that R indicates the set of revoked CHs during the session period l and before, let $R = R_2 \cup \cdots \cup R_{l-1} \cup R_l$, where $|R| \leq t$. In session l, the BS selects a group of users $V_l = \{\text{ID}^l_{\text{CH}_1}, \ldots, \text{ID}^l_{\text{CH}_t}\}$ which meet $\text{ID}_R \subseteq V_l$ and $\text{ID}_{\text{CH}_i} \cap V_l = \emptyset$.

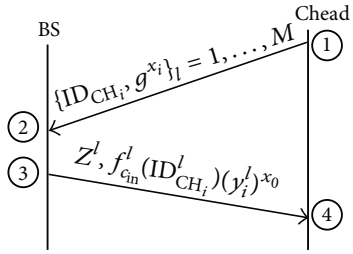
(3) BS broadcasts the information $\{Z^1_{C_{\text{in}}}, Z^2_{C_{\text{in}}}, \ldots Z^l_{C_{\text{in}}}\}$ to each cluster node.

The network key process is shown in Figure 4.
The session key recovery process is as follows.

(1) Having received the key materials, cluster heads calculate their individual share $f^l_{C_{\text{in}}}(\text{ID}^l_{\text{CH}_i}) \cdot (y^l_i)^{x_0} / (y^l_0)^{x_i} = f^l_{C_{\text{in}}}(\text{ID}^L_{\text{CH}_i})$ with the private key x^l_i and public key y^l_0. According to the information $\{\text{ID}^l_i, f_{C_{\text{in}}}(\text{ID}^l_i)\}$, any t sensors or more than t sensor can recover the secret $S^l_{C_{\text{in}}}$ with (7) as follows:

$$f_{C_{\text{in}}}(x) = \sum_{j=1}^{t} \left(\coprod_{l \neq j} \frac{x - \text{ID}^L_{\text{CH}_j}}{\text{ID}^L_{\text{CH}_l} - \text{ID}^L_{\text{CH}_j}} \right) f\left(\text{ID}^L_{\text{CH}_j}\right) \bmod Q,$$

$$S^l_{C_{\text{in}}} = \sum_{j=1}^{t} \left(\coprod_{l \neq j} \frac{\text{ID}^L_{\text{CH}_j}}{\text{ID}^L_{\text{CH}_j} - \text{ID}^L_{\text{CH}_l}} \right) f\left(\text{ID}^L_{\text{CH}_j}\right) \bmod Q. \quad (7)$$



① Chead → BS: $\{ID^L_{CH_i}, g^{x^l_i}\}$

② BS: verify the Chead's $g^{x^l_i}$ and select $f(x)$

③ BS → Cheadi: $\{f^l_{c_{in}}(ID^L_{CH_i})(y^l_i)^{x_0}\}$

CH_i: $Z^l \leftarrow K^l$

④ Chead: compute $f(x)$ with $f^l_{c_{in}}(ID^L_{CH_i})$, then compute S^L, $K^l = Z^l - S^L$

Figure 4: The network in session l.

(2) Using $Z^l_{C_{in}}$ and $S^l_{C_{in}}$, users can get the secret $K^l = Z^l_{C_{in}} - S^l_{C_{in}}$.

4.4. Cluster Key Management. In this phase, the protocol establishes the cluster key between CH and members. Similar to the network key, the cluster key can be generated as follows.

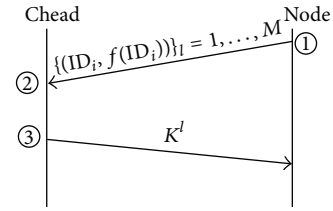
(1) Firstly, cluster head chooses $x^l_{ch_i}$ ($x^l_{ch_i} \in [2, N]$) randomly which relatively primes with $p-1$ and $q-1$, and CH sends it to BS. Then, BS counts out $y^l_{ch_i} = g^{x^l_{ch_i}}$ and sends $(ID^l_{CH_i}, y^l_{CH_i})$ to sensor node in cluster CH_i; meanwhile, sensor node picks $x^l_{i,r}$ randomly which relatively primes with $p-1$ and $q-1$, computes $y^l_{i,r} = g^{x^l_{i,r}} \bmod N$, and then sends $(ID^l_{i,r}, y^l_{i,r})$ $(i = 1, \ldots, m-1, r = 1, \ldots, k)$ to the BS. The BS ensures that if $ID^l_{i,r} \neq ID^l_{i,z}$, there should be no $y^l_{i,r} = y^l_{i,z}$; otherwise it reselects until success.

Furthermore, the BS utilizes CH's $ID^l_{CH_i}$ and members' $ID^l_{i,r}$ $(r = 1, \ldots, k)$ to count out the share $f^l_{CH_i}(ID^l_{CH_i})$ and $f^l_{CH_i}(ID^l_{i,r})$, respectively.

(2) Given that R indicates the set of revoked sensors during the session period l and before, let $R = R_2 \cup \cdots \cup R_{l-1} \cup R_l$, where $|R| \leq t$. During session l, the CH selects a group of users $V_l = \{ID^l_{i,1}, \ldots, ID^l_{i,k}\}$, where $ID_R \subseteq V_l$ and $ID_R \cap V_l = \emptyset$, while BS unicasts $(ID^l_{CH_i}, f^l_{CH_i}(ID^l_{CH_i}) \cdot (y^l_{i,r})^{x^l_{ch_i}} \bmod N)$ to sensor node in CH_i and sends $(ID^l_{i,r}, f^l_{CH_i}(ID^l_{i,r}) \cdot (y^l_{CH_i})^{x^l_{i,r}} \bmod N)$ to CH_i.

(3) Then, BS independently selects M session keys $\{K^l\}_{l=1,\ldots,M}$ from GF(Q), in finite field Q, hiding the K^l: $Z^l_{CH_i} = \{K^l + S^l_{CH_i}\}_{l=1,\ldots,M;\, i=1,\ldots,m}$ with the secret $S^l_{CH_i}$. The purpose is not to leak the K^l. The cluster key process is shown in Figure 5.

4.5. Secret Recovery. Depending on the received information from base station, public generator, node's private key $x^l_{i,r}$, and



① Node: compute $(ID_i, f(ID_i))$

② Chead: compute the Chead's $(ID_i, f(ID_i))$

③ Chead: compute $f(x)$ with $ID^L_{CH_i}, f^l_{c_{in}}(ID^L_{CH_i})$ then compute $f(0) = S^L$, $K^l = Z^l - S^L$

Chead: send K^l to node

Figure 5: The cluster key process.

cluster node's own key $x^l_{ch_i}$, cluster head and members can obtain their share through the following formulas:

$$\frac{f^l_{CH_i}\left(ID^l_{CH_i}\right) \cdot \left(y^l_{i,r}\right)^{x^l_{ch_i}}}{\left(y^l_{ch_i}\right)^{x^l_{i,r}}} = f^l_{CH_i}\left(ID^l_{CH_i}\right),$$
$$\frac{f^l_{CH_i}\left(ID^l_{i,r}\right) \cdot \left(y^l_{ch_i}\right)^{x^l_{i,r}}}{\left(y^l_{i,r}\right)^{x^l_{ch_i}}} = f^l_{CH_i}\left(ID^l_{i,r}\right). \quad (8)$$

$f^l_{CH_i}(ID^l_{CH_i}) \cdot (y^l_{i,r})^{x^l_{ch_i}}$, $f^l_{CH_i}(ID^l_{i,r}) \cdot (y^l_{ch_i})^{x^l_{i,r}}$ is broadcast information; $y^l_{i,r}$ and $y^l_{ch_i}$ are public information. So cluster head and common nodes can obtain their own shares, respectively, and then members send $(ID^l_{i,r}, f^l_{CH_i}(ID^l_{i,r}))$ to CH. CH uses $(ID^l_{CH_i}, f^l_{CH_i}(D^l_{CH_i}))$ and $t-1$ sensors' $(ID^l_{i,r}, f^l_{CH_i}(ID^l_{i,r}))$ to recover the secret S. According to (7), we can carry $S^l_{CH_i}$ out. Furthermore, CH can get $Z^l_{CH_i} = K^l + S^l_{CH_i}$, and then we can calculate K^l to unicast K^l to $t-1$ sensors $(ID^l_{i,r}, K^l)$.

4.6. Scalability. In our solution, we also consider the scalability of network.

4.6.1. New Member Join. When a new member P_u $(u \neq 1, \ldots, n)$ wants to join during session period l, P_u should randomly choose an integer x_u $(x_u \in [2, N])$ and count out $y_u = g^{x_u} \bmod N$. And then, P_u keeps x_u secretly and chooses randomly an ID_u $(ID_u > n)$; n is the largest node identity in network, and then sends (ID_u, y_u) to the BS. The BS will authenticate P_u. If $ID_u > N$ and P_i $(i = 1, \ldots, N)$ and $y_i \neq y_u$, then P_u is acceptable and can join the network.

4.6.2. Node Isolation. Once CH or neighbor nodes find a compromised node P_h, the CH sends its information ID_h to BS. Meanwhile, BS and CH add their IDs into R : $\{R\} \cup \{ID_h\}$.

5. Security Analyses

Due to the unreliable wireless environment, dynamic clustering cluster key distribution scheme is subject to a variety of attacks, such as eavesdropping, tampering, and replay attacks. Compared to previous works, the salient advantage of our solution is that we addressed challenging runtime security issues using localizing key things and group key management based on secret sharing mechanism.

5.1. Robustness. In the recovery phase, for any user $P_i \in P$, if anyone wants to recover K^l, they must obtain both $Z^l_{\mathrm{CH}_i}$ and $S^l_{\mathrm{CH}_i}$, which makes it very difficult to recover keys.

Furthermore, assume that any set $F \subseteq P$ and $|F| \leq t-1$, if an unrevoked user $P_i \in F$, any other user collusions in F cannot get information about the P_i's personal secret S_i. Because in each session l, user P_i's secret $S^l_i = f^l_{\mathrm{CH}_i}(0)$ or $S^l_i = f^l_{\mathrm{C_{in}}}(0)$ is $t-1$ degree polynomial $f^l_{\mathrm{CH}_i}(x)$ or $f^l_{\mathrm{C_{in}}}(x)$, the users in F only know $t-1$ values about $f^l_{\mathrm{CH}_i}(x)$ or $f^l_{\mathrm{C_{in}}}(x)$. And the difficulty to reconstruct a polynomial $f^l_{\mathrm{CH}_i}(x)$ or $f^l_{\mathrm{C_{in}}}(x)$ by $t-1$ values is equivalent to breaking a Shamir's (t, n) secret sharing problem, which is not feasible in computation. Therefore, user collusions in F have no ability to obtain user P_i's secret S^l_i.

Moreover, because the cluster session key K^l is selected from a uniform distribution, and independent of the user's personal secret, no one can obtain information about the session key K^l separated from personal secret collection. Also, in each session $l = 1, \ldots, M$, because of $Z^l_{\mathrm{CH}_i} = K^l + S^l_{\mathrm{CH}_i}$, $Z^l_{\mathrm{CH}_i}$ hides session key K^l with personal secret $S^l_{\mathrm{CH}_i}$, and adversary has no ability to obtain any useful information just from the collection of broadcast messages.

5.2. Tolerance. The normal user $P_i \notin R$ (unrevoked user or normal node) can utilize the broadcast messages and private secret to recover the session key K^l; however, the revoked user can only obtain $t-1$ values from the broadcasted polynomial $f^l(x)$; thus, they have no ability to reconstruct $t-1$ degree polynomial $f^l(x)$ as mentioned above. Therefore, the user in R cannot get $f^l(0)$. Moreover, because of $K^l = Z^l_{\mathrm{CH}_i} - f^l(0)$, it is not feasible to recover personal secret K^l by $Z^l_{\mathrm{CH}_i}$ and $S_{r\{p_r \in R\}}$.

5.3. Security. Our solution also has both $t-1$ forward secrecy and $t-1$ backward secrecy.

5.3.1. $t-1$ Forward Secrecy. Let $R \subseteq P$, $|R| \leq t-1$, and each $P_i \in R$ is a revoked user before session l. Even if user collusions know all cluster keys before the session l, they cannot obtain any information of current session key K^l, because they cannot recover $f^l(0)$ with just $t-1$ values of $f^l(x)$. Therefore, the solution is $t-1$ forward secrecy.

5.3.2. $t-1$ Backward Secrecy. Let $J \subset P$, $|J| \leq t-1$. Each user $P_i \in J$ joined the group before session l. Even if user collusions in J know all cluster keys K^l before the session l, they cannot obtain any information of current session key K^{l_1} ($l_1 < l$). Because if a user wants to get K^{l_1}, the user $P_i \in J$ at least recovers t points of $f^{l_1}(x)$ for $f^{l_1}(0)$. However, each user P_i after session J at least obtains $t-1$ value from $t-1$ degree $f^{l_1}(x)$ and has no ability to reconstruct $f^{l_1}(x)$; that is, the solution is $t-1$ backward secrecy.

5.4. Complexity Analysis. In this section, we discuss the complexity of our scheme from computation complexity, communication complexity, and storage cost needed by common node and cluster.

(1) Computation complexity: we assume that base station has a large computation capacity, the pickout of polynomial and share distribution as well as the choice of generator. Common node only needs to compute division, and cluster head needs to reconstruct the polynomial beside division.

(2) Communication costs include the broadcast cost: $j(\log^q)$ and download the publishing information from publish board: $t * j * (\log^q)$, where t indicates a session period; j is the number of nodes in one cluster, and q is an enough secure prime.

(3) Storage cost: in our scheme, we only need to save a private respective key, which is $\log^q$.

6. Conclusion and Future Work

In this paper, we propose a secret sharing-based key management scheme (SSKM) to enhance network security and survivability. Different from previous works, although we employ the hierarchical architecture, we limited the size of clusters to balance the overall energy consumption of the network. In contrast to other clustered architectural security solutions, the salient advantage of this work is that we addressed challenging security issues by localizing key things based on secret sharing theory. We present the network key and cluster key and generate new keys from various polynomials by Lagrange interpolation formula. Also, we present a rekey mechanism in the cluster head selection with low energy consumption. Meanwhile, SSKM has an authentication mechanism to ensure the scalability, which cannot only authenticate the new sensor but also can isolate the compromised node. The security analysis shows that our solution cannot only reduce the energy consumption effectively but also enhance the security level. In the future, we will focus on how to enhance security in mobile and scalable WSNs.

Acknowledgments

This work was supported by China Postdoctoral Science Foundation Funded Project (2012M510367); the National Basic Research Program of China (973 Program) (2011CB302900).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] Y. Zhou, Y. Fang, and Y. Zhang, "Securing wireless sensor networks: a survey," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 3, pp. 6–28, 2008.

[3] K. Vivek, C. Narottam, and S. Surender, "A survey on clustering algorithms for heterogeneous wireless sensor networks," *International Journal of Advanced Networking and Applications*, vol. 2, no. 4, pp. 745–754, 2011.

[4] X. Du, Y. Xiao, M. Guizani, and H. H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 24–34, 2007.

[5] M. Bertier, A. Mostefaoui, and G. Trédan, "Low-cost secret-sharing in sensor networks," in *Proceedings of the IEEE 12th International Symposium on High Assurance Systems Engineering (HASE '10)*, pp. 1–9, November 2010.

[6] T. Claveirole, M. Dias De Amorim, M. Abdalla, and Y. Viniotis, "Securing wireless sensor networks against aggregator compromises," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 134–141, 2008.

[7] H. N. Seyed, H. J. Amir, and D. Vanesa, "A distributed group rekeying scheme for wireless sensor networks," in *Proceedings of the 6th International Conference on Systems and Networks Communications (ICSNC '11)*, pp. 127–135, 2011.

[8] Y. Y. Zhang, X. Z. Li, J. M. Liu, J. C. Yang, and B. J. Cui, "A secure hierarchical key management scheme in wireless sensor network," *The International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 547471, 8 pages, 2012.

[9] Y. Y. Zhang, X. Z. Li, J. C. Yang, Y. A. Liu, N. X. Xiong, and A. V. Vasilakos, "A real-time dynamic key management for hierarchical wireless multimedia sensor network," *Multimedia Tools and Applications*, 2012.

[10] Y. Y. Zhang, W. C. Yang, K. B. Kim, and M. S. Park, "An AVL tree-based dynamic key management in hierarchical wireless sensor network," in *Proceedings of the 4th International Conference on Intelligent Information Hiding and Multiedia Signal Processing (IIH-MSP '08)*, pp. 298–303, August 2008.

[11] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: a secure sensor network communication architecture," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 479–488, April 2007.

[12] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: efficient security mechanisms for large-scale distributed sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 500–528, 2006.

[13] M. Eltoweissy, M. Moharrum, and R. Mukkamala, "Dynamic key management in sensor networks," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 122–130, 2006.

[14] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[15] G. Blakley, "Safeguarding cryptgraphic keys," in *Proceedings of the National Computer Conference (AFIPS '79)*, pp. 313–317, AFIPS Press, New York, NY, USA, 1979.

[16] S. Agrawal, "Verifiable secret sharing in a total of three rounds," *Information Processing Letters*, vol. 112, pp. 856–859, 2012.

[17] C. Hua, X. Liao, and X. Cheng, "Verifiable multi-secret sharing based on LFSR sequences," *Theoretical Computer Science*, vol. 445, pp. 52–62, 2012.

[18] Y.-X. Liu, L. Harn, C.-N. Yang, and Y.-Q. Zhang, "Efficient (n, t, n) secret sharing schemes," *Journal of Systems and Software*, vol. 85, no. 6, pp. 1325–1332, 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 390194, 5 pages
http://dx.doi.org/10.1155/2013/390194

Research Article

An Improved RFID-Based Localization Algorithm for Wireless Sensor Networks

Ayong Ye, Yuanjing Ling, Li Xu, and Xiaoliang Yang

Key Lab of Network Security and Cryptology, Fujian Normal University, Fuzhou 350007, China

Correspondence should be addressed to Ayong Ye; yay@fjnu.edu.cn

Received 25 January 2013; Accepted 14 April 2013

Academic Editor: Muhammad Khurram Khan

The position of RFID-tagged objects in sensor networks is becoming an important feature for ubiquitous computing applications. In this work, we develop an improved positioning algorithm for RFID-tagged objects. Compared with the VIRE algorithm, we apply a nonlinear interpolation method to further improve the positioning precision. We also introduce the concept of credibility of reference tags to reduce the influence of abnormal tags on positioning precision. Finally, we provide detailed simulations to evaluate the proposed technique.

1. Introduction

Location sensing is a crucial prerequisite for many wireless sensor networks (WSNs) applications [1] such as asset tracking, industrial automation, and homecare and healthcare systems. GPS is the most widely used technology for localization. However, once arriving at the earth surface, GPS signals are too weak to penetrate the building. To overcome those limitations, various localization technologies have been proposed for WSNs among which are indoor GPS, motion and rotation sensors, infrared, ultrasound, ultra-wideband (UWB), wireless local area network (WLAN), and radio frequency identification (RFID). Due to its lower cost and its technical capabilities, RFID has been widely adopted as an attractive technology for many WSNs' applications, which uses wireless communication for automatic identification of objects.

In the indoor environment, layout structure and moving objects can cause reflection, refraction, diffraction, dead spots, and absorption of radio signals. Therefore, the multipath phenomenon will occur and affect the accuracy of indoor location. Moreover, many other factors also influence the RSSI, such as temperature, orientation of antenna, and height to the ground. LANDMARC [2] was the first attempt using active RFID for indoor location sensing, while VIRE [3] improved it by introducing the concepts of virtual reference tags and proximity map. In this work, we develop an improved positioning algorithm for RFID-tagged objects, which not only achieves robustness to abnormal tags but also dramatically improves the positioning precision.

The rest of the paper is organized as follows. The next section presents a brief review on the related work. Our new approach, NICoT (RFID localization algorithm based on nonlinear interpolation and credibility of tags), is detailed in Section 3. We respectively evaluate the performance of our scheme by simulations in Section 4. Finally, we conclude this paper in Section 5.

2. Related Work

RFID-based ILS solutions can be categorized into the following three areas based on their localization methods: triangulation-based solutions, proximity, map-based solutions, and scene analysis solutions.

(1) Triangulation [4, 5]: the distance between the reader and tag is calculated by using the collected signal strength, and the distances are used for triangulation to obtain the position of the tracking tag. The triangulation method depends largely on an accurate signal propagation model, which greatly decreases its effectiveness.

(2) Scene analysis [6–10]: the scene analysis method captures the signal strength in the sensing area and compares it with a preexisting signal strength database

to map the target to its location. The scene analysis method needs to build a radio map by calibrating signal strength values at training locations, which is used repeatedly afterwards. However, the signal strength values change with the change of the environment. The fingerprint has to be updated periodically, which requires considerable time and labor.

(3) Proximity [6, 11, 12]: the proximity method needs to measure the nearness of a set of neighboring reference tags. Those tags are close to the target, and their locations are fixed and known. The measured nearness, along with the corresponding known locations, is used to estimate the location of a target. The proximity map is a promising method for complex and dynamic built environments. This method can accommodate the impacts of the complexity and dynamic nature of built environments. It does not use the RSSI value to estimate the travel distance of the signal. Instead, it uses RSSI value to compare the nearness of reference tags to the target, which is qualitative and less sensitive to the obstructions that impact the propagation of the signal.

In LANDMARC, the concept of reference tags is introduced, which can provide reference locations. The known locations of the nearest neighboring reference tags and nearness to the tracking tag are used in computing the tracking tag's location. The algorithm has the following advantages. First, the algorithm uses tags instead of more readers, which greatly reduces the cost of system. Second, the reference tags and target tags are in the same environment, and the effect of environmental factors can be effectively off-settled. However, the algorithm has the following drawbacks. First, it does not work well in a closed area with severe radio signal multipath effects. Second, to further improve the localization accuracy, more reference tags are needed, which is costly and may cause RF interference phenomenon. In VIRE, the virtual reference tags were used to achieve a higher accuracy. These tags are virtually distributed linearly between real reference tags, which increase the density of the reference tag grids. Locations of the virtual reference tags are known and recorded, and their signal strength is estimated by linear interpolation of that of the real tags next to them. The sensing area is divided into small regions, and each reader maintains its own proximity map. As a result, the most probable location can be estimated.

3. The NICoT Approach

3.1. Credibility of Real Reference Tags. Due to hardware faults or external interference, the signal strength of tags may be erroneously read by RF reader. In the practical measurement, we observe that some tags' signal strengths read by readers are too large or too small, or zero (fail to read). Therefore, the signal strength of the reference tags is not trustworthy and needs to be verified before they are chosen for the tracking tag's location calibration.

With the placement of the reference tags and the readers shown in Figure 1, we can observe that a reference tag's signal strength is always similar to its neighbor's for the same reader.

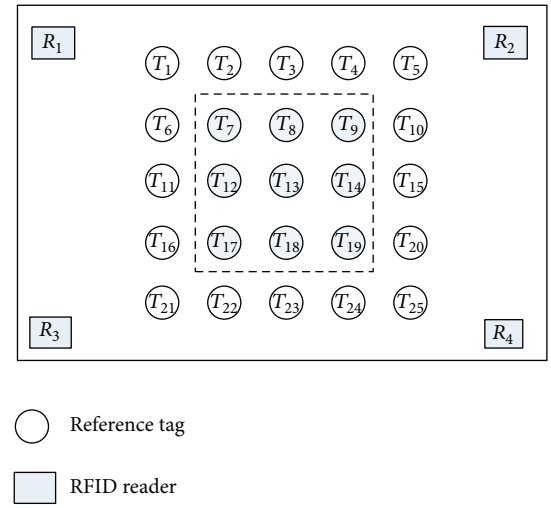


Figure 1: Placement of RF readers and tags.

Therefore, with known signal strength of all reference tags, we can check whether the measurement is proper or not. Suppose we have n RF readers $(R_1, R_2, \ldots, R_n)$ along with m tags $(T_1, T_2, \ldots, T_m)$ as reference tags. We define the signal strength of all reference tags perceived on all readers as an $n \times m$ matrix SSM, where SSM (i, j) denotes T_i's signal strength perceived on R_j. We also define the difference of signal strength between T_i and its neighborhood as an $n \times m$ matrix DSSM, where DSSM (i, j) denotes whether the T_i's signal strength is similar to its neighborhood's average signal strength or not. For the same reader R_j, if the difference between T_i's signal strength and its neighborhood's average signal strength is less than the threshold τ, then the element at row i and column j will be 1. Otherwise, it will be 0; namely,

$$\mathrm{DSSM}(i, j) = \begin{cases} 1, & \text{if } \left|\mathrm{SSM}(i, j) - \mathrm{average}_{T_x \in N}\left(\mathrm{SSM}(x, j)\right)\right| \le \tau \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Here, N is the set of T_i's neighbors. Empirically, the threshold τ can be calculated as follows:

$$\tau = \max_{T_x \in N}\left(\mathrm{SSM}(x, j)\right) - \frac{\min_{T_x \in N}\left(\mathrm{SSM}(x, j)\right)}{2}. \tag{2}$$

As an example in Figure 1, the set of T_{13}'s neighbors is $\{T_7, T_8, T_9, T_{12}, T_{14}, T_{17}, T_{18}, T_{19}\}$. Therefore, the credibility $c(T_i)$ of the tag T_i can be computed as follows:

$$c(i) = \frac{\sum_{j=1}^{n} \mathrm{DSSM}(i, j)}{n}, \tag{3}$$

where $0 \le c(i) \le 1$, and n is the number of all RF readers.

3.2. Nonlinear Interpolations. The concept of virtual reference tags can increase the precision of position estimation without additional cost. However, the VIRE algorithm calculates the virtual tags' signal strength by a simple linear

method, which does deteriorate the performance of interpolation. To further improve the positioning precision, we apply a simple nonlinear interpolation in the solution model.

As shown in Figure 2, the real reference tags in our approach are also properly placed to form a 2D regular grid. Each physical grid cell covered by 4 real tags is further divided into $n \times n$ equal virtual grid cells. Therefore, each virtual grid cell can be considered as covered by four virtual reference tags. Since the coordinates of the four real reference tags are known, the coordinates of the virtual reference tags can be easily calculated.

The RSSI values of those virtual tags are a function related to the signal strength of 4 real reference tags in corner and can be obtained by the empirical formulas shown below:

$$S_k\left(T_{i,j}\right) = \left(S_k\left(T_{0,0}\right) \times D_a^2 + S_k\left(T_{0,1}\right) \times D_b^2 + S_k\left(T_{n,0}\right) \times D_c^2 + S_k\left(T_{n,n}\right) \times D_d^2\right) \times \left(D_a^2 + D_b^2 + D_c^2 + D_d^2\right)^{-1}. \quad (4)$$

Here, $S_k(T_{i,j})$ represents the RSSI value of the reference tag located at the coordinate (i, j) for the kth reader. D_a, D_b, D_c, and D_d denote, respectively, the distance between the virtual reference tag T_{ij} and the four real reference tags $T_{0,0}$, $T_{0,n}$, $T_{n,0}$, and $T_{n,n}$.

3.3. Weight of Reference Tags. The proximity map is also used to eliminate those unlikely positions in the NICoT. When the interpolation and elimination processes are finished, we can obtain a set of possible positions. To improve the accuracy and robustness of NICoT, one weighting factor w_i' is introduced. The weighting factor w_i' not only demonstrates the discrepancy of RSSI values between the selected virtual reference tags and the tracking tag but also indicates the credibility of the reference tags. We observe that the reference information of a closer tag is more accurate and trustable, which gives a larger RSSI value. The w_i' is a function of the RSSI values of selected virtual tags, as given by

$$w_i' = \frac{1/E_i^2}{\sum_{j=1}^{k}\left(1/E_j^2\right)}. \quad (5)$$

Here, E_i is the Euclidean distance in signal strength between a tracking tag and a virtual reference tag T_i. We define the signal strength vector of a tracking tag as $S = (S_1, S_2, \ldots, S_n)$, where S_i denotes the signal strength of the tracking tag perceived on reader R_i. For the reference tags, we denote the corresponding signal strength vector as $\theta = (\theta_1, \theta_2, \ldots, \theta_n)$, where θ_i denotes the signal strength. Then, E_i is given by

$$E_i = \sqrt{\sum_{p=1}^{n} \frac{\left(\theta_p - S_p\right)^2}{\theta_p}}. \quad (6)$$

When we compute the coordinate of a tracking tag, the weighting factors should be considered. We suggest an assorted weight w_i as $w_i = w_i' \times C(T_i)$. The calculated coordinate is given by

$$(x, y) = \sum_{i=1}^{k} w_i \times (x_i, y_i). \quad (7)$$

Here, k is the number of total regions in the set of possible positions and (x_i, y_i) denotes the coordinates of the virtual reference tag T_i.

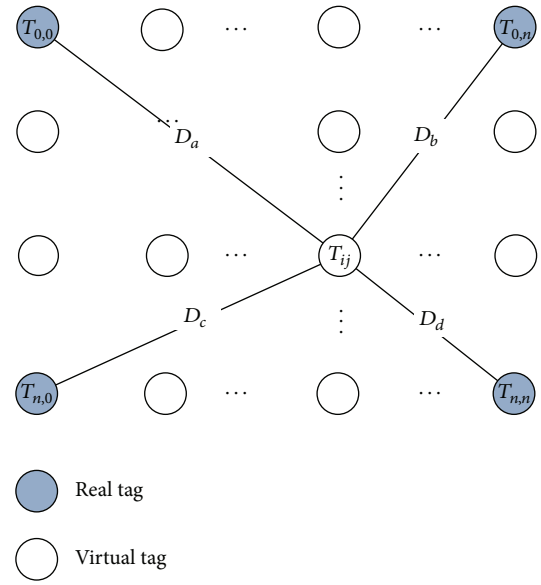


Figure 2: Nonlinear interpolation.

4. System Implementation and Simulation Evaluation

4.1. System Implementation. We use the simulation programs run in MATLAB to evaluate the performance of NICoT in an idealized network environment. The experiment environment is illustrated in Figure 3. In our environment, 16 real reference tags were set in 4 m ∗ 4 m area; the distance between them was 1 m; 4 readers were set in each corner; and 9 tracking tags were set too. Virtual reference tags were inserted as in Figure 2. We define the location estimation error e to be the linear distance between the tracking tag's real coordinates (x, y) and the computed coordinates (x', y'), given by $e = \sqrt{(x - x')^2 + (y - y')^2}$.

4.2. Effect of the Density of Virtual Reference Tags. Figure 4 shows the effect of the number of virtual reference tags on the estimation error when $\tau = 2$. We can see that the higher the density of virtual reference tags, the greater the location sensing precision. However, the positioning accuracy will not improve as the number of virtual reference tags is beyond a certain number. We note that the accuracy does not improve significantly when the number of reference tags increases from 30×30 to 40×40. In our experiments, each physical grid cell covered by 4 real tags is further divided into 30×30 equal virtual grid cells.

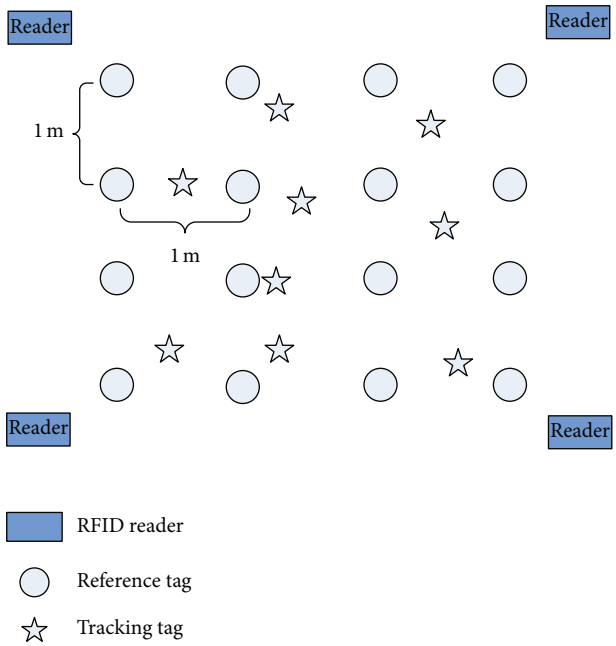


Figure 3: Experiment environment.

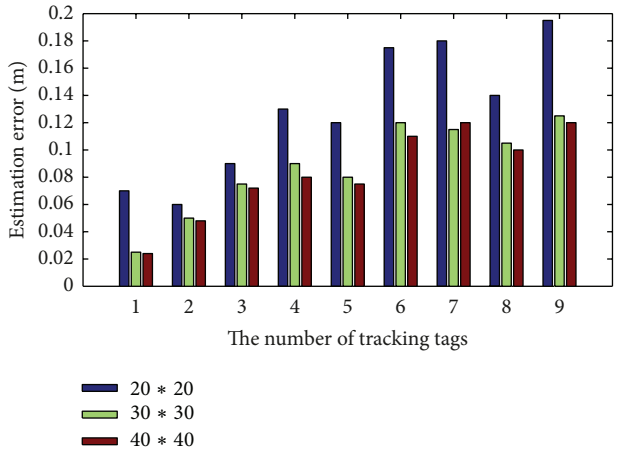


Figure 4: Estimation error of the 9 tracking tags.

4.3. Performance Comparison Using Different Parameters. Figure 5 shows that different threshold τ has different estimation error. We can see that the cumulation of estimation error under different choices of τ increases with the increase of error. We also note the estimation error is minimum, when $\tau = 2$ is chosen.

4.4. Performance Comparison Using Different Localization Algorithms. Figure 6 shows the comparison between the NICoT approach and the VIRE approach for 9 different locations of the tracking tags. The reduction in estimation error for NICoT is from 43.6% to 62.8% over VIRE for all 9 locations. For those tags, the worst estimation error is 0.228 m and the average estimation error is 0.14 m. Clearly, the NICoT approach provides a higher degree of accuracy than that of VIRE at all locations.

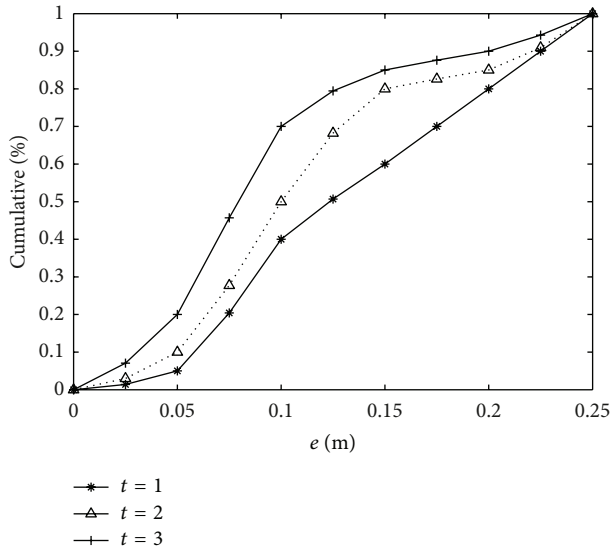


Figure 5: Estimation error of the 9 tracking tags.

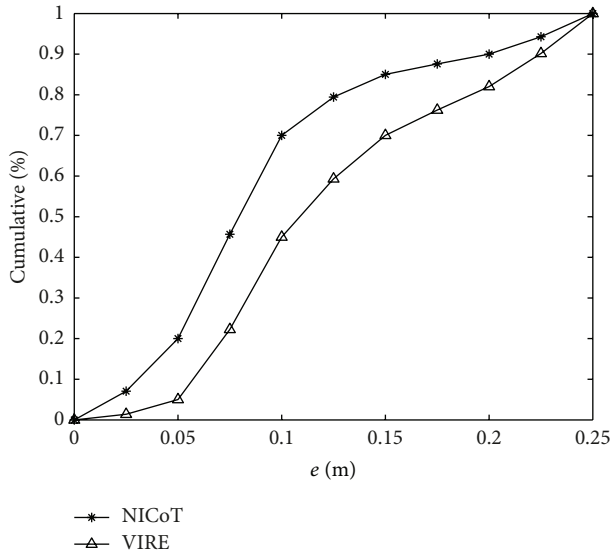


Figure 6: Cumulative percentile of error distance.

5. Conclusion and Future Work

This paper presented an improved algorithm based on nonlinear interpolation of the virtual reference tags' signal strength and the concept of the reference tags' credibility. Compared with the VIRE algorithm, we apply a nonlinear interpolation method to further improve the positioning precision. We also introduce the concept of credibility of reference tags to reduce the influence of abnormal tags on positioning precision. It can be used for real-time indoor locating. Obviously, the proposed algorithm can be easily extended to the three-dimensional space. In future research, we may combine our algorithm with other location algorithms to further improve the positioning accuracy.

Acknowledgments

This work was supported by NSF of China under Grant nos. 61202452, 61202450, and 61202451 and the CEEUSRO projects of Fujian Province Education Department (no. JA12063).

References

[1] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of "two-factor user authentication in wireless sensor networks"," *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.

[2] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: indoor location sensing using active RFID," in *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications*, pp. 407–415, 2003.

[3] Y. Zhao, Y. Liu, and L. M. Ni, "VIRE: Active RFID-based localization using virtual reference elimination," in *Proceedings of the International Conference on Parallel Processing (ICPP '07)*, p. 56, 2007.

[4] J. Hightower, G. Boriello, and R. Want, "SpotON: an indoor 3D location sensing technology based on RF signal strength," Tech. Rep. UW CSE 00-02-02, Department of Computer Science and Engineering, University of Washington, Seattle, Wash, USA, 2000.

[5] X. Luo, W. J. O'Brien, and C. L. Julien, "Comparative evaluation of Received Signal-Strength Index (RSSI) based indoor localization techniques for construction jobsites," *Advanced Engineering Informatics*, vol. 25, no. 2, pp. 355–363, 2011.

[6] N. Li and B. Becerik-Gerber, "Performance-based evaluation of RFID-based indoor location sensing solutions for the built environment," *Advanced Engineering Informatics*, vol. 25, no. 3, pp. 535–546, 2011.

[7] Z. N. Zhen, Q. S. Jia, C. Song, and X. Guan, "An indoor localization algorithm for lighting control using RFID," in *Proceedings of the IEEE Energy 2030 Conference (ENERGY '08)*, pp. 1–6, Atlanta, Ga, USA, November 2008.

[8] A. Pradhan, E. Ergen, and B. Akinci, "Technological assessment of radio frequency identification technology for indoor localization," *Journal of Computing in Civil Engineering*, vol. 23, no. 4, pp. 230–238, 2009.

[9] A. Papapostolou and H. Chaouchi, "RFID-assisted indoor localization and the impact of interference on its performance," *Journal of Network and Computer Applications*, vol. 34, no. 3, pp. 902–913, 2011.

[10] W. S. Ku, K. Sakai, and M. T. Sun, "The optimal k-covering tag deployment for RFID-based localization," *Journal of Network and Computer Applications*, vol. 34, no. 3, pp. 914–924, 2011.

[11] Y. Huang, Z. Lui, and G. Ling, "An improved Bayesian-based RFID indoor location algorithm," in *Proceedings of the International Conference on Computer Science and Software Engineering (CSSE '08)*, pp. 511–514, December 2008.

[12] Y. Huang, L. V. Shilei, H. E. Yuwei, and H. Jianfeng, "An isosceles triangular placement of reference tags for RFID indoor location system," *Chinese Journal of Electronics*, vol. 20, no. 3, pp. 504–510, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 582656, 6 pages
http://dx.doi.org/10.1155/2013/582656

Research Article

A Dynamic and Distributed Scheduling for Data Aggregation in Ubiquitous Sensor Networks Using Power Control

S. Madhavi[1] and Tai Honn Kim[2]

[1] *Department of Computer Science and Engineering, P.V.P. Siddhartha Institute of Technology, Andhra Pradesh, India*
[2] *Department of Convergence Security, Sungshin Women's University, 249-1 Dongseon-dong 3-ga, Seoul 136-742, Republic of Korea*

Correspondence should be addressed to Tai Honn Kim; taihonn@daum.net

Received 28 January 2013; Accepted 19 April 2013

Academic Editor: Muhammad Khurram Khan

Data aggregation scheduling in ubiquitous sensor networks is a major research interest for many researchers. Very little research is carried out to schedule the nodes in ubiquitous sensor networks for maximizing the throughput and utilize the hardware resources effectively. The traditional graph model does not model the interferences occurring from the concurrent transmissions in the Ubiquitous Sensor Networks. Hence in this paper we propose a new network model for the USN called a power control collision interference free model and a novel distributed data aggregation scheduling protocol which is adaptive to the rate and power. We tested the protocol in a USN consisting of 50, 100, 150, 200, and 250 number of nodes with different topologies and varying degree 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, . . ., 40. We compared our results with the other protocols and proved that our proposed power control collision interference free model yields better results.

1. Introduction

RFID (radio frequency identification) is a technology which is used to acquire information anytime and anywhere through network access service [1]. Such ubiquitous network yields better results when integrated with wireless sensor networks (WSNs). This integrated network is called a ubiquitous sensor network (USN) [2]. Several servers participate in ubiquitous networks and the flow of data across the network is as follows

(1) the sensors detect events,

(2) the RFID readers recognize tags,

(3) the information is forwarded to one or more servers,

(4) depending on the arrived event, the servers integrate data from RFID networks and WSNs fire the necessary procedures,

(5) each intermediate sensor implements method for data aggregation and filtering.

Generally the following are some of the limitations of RFID networks when integrated with WSN:

(1) all the nodes share a single wireless channel for transmission and when multiple nodes transmit at the same time, their transmission may get collided,

(2) nodes cannot receive and send simultaneously since they are equipped with a single half-duplex radio transceiver [3],

(3) bandwidth and battery power are the two resource limitations for RFID active tags and sensors,

(4) retransmissions after collisions, overhearing and overemitting reduce the efficiency of USN.

Hence in order to overcome such limitations, an efficient schedule for data aggregation is needed. Each ubiquitous node in USN should forward data to the base station. Each node does not forward data to the base station directly. Instead it forwards data to a special node called head node. The head node in turn forwards data to the base station.

Each node obtains a frequency for transmitting in one or more time slots.

The following are the two main objectives of our paper:

(1) selection of the head node,

(2) finding a schedule which decides the following:

  (a) when head nodes should transmit,
  (b) how the head nodes gather data from their neighbors.

Since the ubiquitous nodes change their state dynamically, finding a solution to the above mentioned tasks is very difficult.

An optimal schedule is a schedule with

(a) minimum number of time slots (frame length),

(b) maximum number of concurrent transmissions per each time slot.

USNs are commonly modeled using

(1) traditional graph model,

(2) physical interference model.

The traditional graph model does not model the interferences occurring from the concurrent transmissions.

In the physical interference model, the transmission becomes successful if and only if the signal-to-interference-noise-ratio (SINR) perceived at the receiver is above a certain threshold.

For example, if V_j denotes a node that can transmit concurrently with node V_i as the head, then for a successful transmission the SINR perceived at the receiver should be greater than or equal to β.

That is,

$$\frac{P_s/d_s^{\alpha}}{N^0 + \sum_{j=1}^{\text{conc}} \left(P_r/d_j^{\alpha}\right)} \geq \beta, \tag{1}$$

where

(a) P_s and P_r are the transmission powers of sender and receiver nodes,

(b) d_s is the distance between the sender and receiver nodes,

(c) d_j is the distance between a concurrent transmitter and the receiver nodes,

(d) α is the path loss ratio which has a typical value between 2 and 4,

(e) N^0 is the ambient noise, and

(f) β is the threshold for a successful transmission.

Earlier several centralized and distributed scheduling algorithms are proposed for data aggregation in USN [4, 5]. In this paper, we presented a novel distributed data aggregation scheduling protocol using a power control collision-interference-free model. Our proposed protocol is adaptive to the rate and power control and also yields better results when compared with the other existing protocols.

2. The Proposed Power Control Collision Interference Free Ubiquitous Sensor Network Model

Consider a ubiquitous sensor network with n arbitrarily distributed ubiquitous nodes. Let a directed graph $G = (V, E)$ denote a ubiquitous network, where $V = \{v_0, v_1, v_{n-1}\}$ and $E = \{e_0, e_1, e_{n-1}\}$. V_i denotes ubiquitous node i, and e_i denotes the edge between two ubiquitous nodes. The minimum Euclidean distance between a pair of nodes is 1. The transmission is assumed to be half duplex.

Each node in the proposed power control collision interference free model is characterized by an 8-variable vector

(
Nodetype Nodeid
Info Location
NodeState State
Integer number of neighbors
CRange communication range, Max Crange
IRange interference range, Max Irange
Power transmission power, MaxPower
Distance distance from base station
)

The location denotes the global location of a ubiquitous node. Each node exists in two states called active and inactive states. The nodes that are willing to transmit are said to be in active state and are termed as active nodes, while the remaining nodes are said to be in inactive state and termed as inactive nodes. The state of a node changes frequently.

Now we model a ubiquitous sensor network as USN (V, crmax, ρ), where the following hold.

(1) V is the set of n ubiquitous nodes each characterized with the 8-variable vector.

(2) crmax is the maximum communication range.

(3) $\rho \geq 1$ is the interference factor.

(4) Each node can send and receive data in the maximum range crmax.

(5) The transmission power of a node is always $\leq$ crmax.

(6) Let V_j denote a node that can transmit concurrently with node V_i as the head. For a successful transmission the SINR perceived at the receiver should be greater than or equal to β; that is,

$$\left(\frac{\left(P_s/d_s^{\alpha}\right)}{\left(N^0 + \sum_{j=1}^{\text{conc}} \left(P_r/d_j^{\alpha}\right)\right)}\right) \geq \beta. \tag{2}$$

Let P_s and P_r denote the transmission powers of sender and receiver nodes respectively, d_s is the distance between sender and receiver nodes and d_j is the distance between a concurrent transmitter and the receiver, α is the path loss ratio, which has a typical value between 2 and 4, N^0 is the ambient noise, and β is the threshold for a successful transmission.

(7) Every node has the flexibility to tune itself to an optimal power.

The spatial reuse and data rates are the two important factors to be considered to improve the efficiency of the USN. As the signal strength decreases with the square of the distance according to the free space path loss model, the interference range is assumed to be two times the required distance between the nodes in order to avoid interferences. Generally larger transmission powers result in high data rates but allow a few concurrent transmitters. Similarly smaller transmission powers result in low data rates but allow more concurrent transmitters. Hence an optimal transmission powers are chosen to improve the overall network capacity.

3. Distributed Algorithm

The models for representing Networks are divided into three categories [6]:

(1) tree based,

(2) cluster based,

(3) hybrid topology.

In tree based structure, the root of tree is called sink node. Data flows from leaf nodes to sink nodes through one or more intermediate nodes [6, 7]. These intermediate nodes are called aggregator nodes. Aggregator nodes in turn add their own monitored data and send the aggregated data to its parent node [6].

In clustered approach, every cluster has a cluster head and the entire node in this cluster transmit data to the cluster head (CH). This CH either transmits data to sink node or to the CH of another cluster which is nearest to sink node [6].

In [8] a new adaptive distributed broadcast scheduling method for mobile ad hoc networks is proposed. In [9, 10] a method for maximizing the channel throughput in mobile ad hoc networks using mIDS is proposed. In [11] an ACSP interceptor for improving network capacity in mobile ad hoc networks is proposed.

In this paper we proposed a novel solution for improving the performance of USN. It consists of the following three steps:

(1) identification of ConcHead Nodes,

(2) constructing a schedule with a list of neighbors of ConcHeads,

(3) finding an optimal schedule.

The protocol Optimal_SHDL will

(1) identify the ConCHead nodes,

(2) construct a set of schedules from which an optimal schedule is also founded.

Generally time is slotted to intervals where L denotes the length of the interval.

A schedule is a sequence of time slots. All noninterfereing nodes u_i are scheduled to send their aggregated data to any of their neighbors, v_i. The power p_i chosen at v_i cannot exceed the value of the MaxPower. Formally, at each time slot t, we have an assignment vector:

$$\text{AssignVect}_k = \left((u_1, v_1, p_1), (u_2, v_2, p_2), \ldots, (u_k, v_k, p_k)\right). \tag{3}$$

Generally if each node knows when to transmit, ideal listening can be avoided. Every node after receiving or transmitting data goes to sleep mode or for processing the data until its next time slot, and this technique saves battery power [6].

Our distributed algorithm is an iterative process and ends when the complexity of the ConCHeadTree cannot be further decreased. The algorithm is as follows.

Algorithm 1.0 PCCIF (UV, N)

UV represents the set of of Ubiquitous nodes
N represents the total number of nodes

Step 1. Construct a PCCIF_Network from the set of nodes UV

Step 2. ConCHeadTree = Tree_traversal (PCCIF_Network, UV, N)
//Construct ConCHead Tree from the PCCIF_Network

Step 3. ConCHeadList = Optimal_SHDL (ConCHeadTree)
//Construction of Optimal Schedule contains three phases where we identify a list of ConCHeadnode, their neighbors, the schedule for transmission and an optimal schedule

Step 4. Empty List UV

Step 5. Set UV = ConCHeadList
//We rebuild a USN with the ConCHeadnode identified and in the new USN, the ConCHeadnode are the normal nodes

Step 6. Find complexity of the new USN

Step 7. If complexity level is above a threshold then
Forward_to_Base(ConcHeadList)
Goto Step 8
Else
Goto Step 1
Step 8. End.

An optimal schedule is constructed with edges from E. The total transmission time T consists of slots $\{t_0, \ldots, t_{T-1}\}$ and the total schedule $S = \{S_0, S_1, \ldots, S_{T-1}\}$, where S_t denotes the subset of nodes in V scheduled to transmit in time slot t_i.

The optimal schedule satisfies the following conditions:

(a) each active node may be scheduled more than once but must be scheduled at least once,

(b) a node cannot act as a transmitter and a receiver in the same time slot, in order to avoid the primary interference [12],

(c) a head node vi transmits to the base station only after all its neighbors have been scheduled [13],

(d) we use the path loss radio propagation model [14] to characterize path loss:

$$p_{\text{recv}} = \frac{p_{\text{trans}}}{d^{\alpha}}, \tag{4}$$

where p_{recv} and p_{trans} are the transmit powers at the receiver and transmitter, d is the distance between the transmitter and the receiver, and α is the path loss exponent, ranging from 2 (line of sight free space) to 4 (indoor).

The construction of an optimal schedule from the proposed power control collision interference free model is as follows.

Given a USN tree, then the procedure *Optimal_SHDL (ConCHeadTree)* consists of the following phases.

(1) Head Selection Phase. Construct one or more ConCHeadTrees and from each ConCHeadTree identify one or more ConCHeads.

(2) Aggregation Scheduling Phase. Construct a schedule with the list of active nodes for each ConCHeadnode. All the neighbors of each ConCHeadnode are assigned different frequencies from the Frequencyset $\{f_1, f_2, f_3, f_4, \ldots, f_n\}$. Hence they can transmit to the ConCHeadnode at the same time concurrently in the same slot without any interference.

Let AV_{ij}, for $i, j = 1, \ldots, n$, denote the set of active nodes for each ConCHeadnode_i. Assign a unique frequency and a time slot in the schedule for each AV_{ij}. These frequencies are allocated in such a way that there are no interferences among the concurrent transmitters and the SINR constraint is satisfied.

The above two steps are iterative and end when all the nodes are termed as either ConCHeadnodes or a neighbor to the ConCHeadnodes. There exists one or more ConCHeadnodes to forward the aggregated data to the base station and each ConCHeadnode has zero, one or more neighbors from where it gathers the data. Generally a ConCHeadnode is selected based on a criterion that improves the utilization of the hardware resources and maximizes the network throughput. The schedule consists of one or more slots. And at the end of the two phases a set of schedules are identified.

(3) Optimal Scheduling Phase. The optimal schedule is constructed from the schedules constructed in the previous phases. An optimal schedule always improves the network capacity and also best utilizes the hardware resources at each ubiquitous node.

Algorithm 2.0 Optimal_*SHDL*

Step 1. Let N denote the number of nodes in USN and UV denote the set of all N nodes in USN

Step 2. for $i = 1, 2, 3, \ldots, N$ do Steps 3 to Steps 6

Step 3. NodeSet = UV, $j = 1$

Step 4. Increment s by 1

Step 5. Repeat until NodeSet = NULL or no new traversal on USN

(5.1) Increment k by 1

(5.2) Empty ConCHeadList

(5.3) {ConCHeadList[UV_i], Totalnodes[$\text{UV}_{i,j}$]} = Tree_Traversals(UV_i, i, j, UV)

(5.4) Construct Schedule [s, Slot[k], ConCHeadList [UV_i], totalnodes [$\text{UV}_{i,j}$]]

(5.5) For each node X in ConCHeadList do NodeSet = NodeSet − Activeneighbor[X]

(5.6) NodeSet = NodeSet – ConCHeadList

(5.7) Increment j by 1

End

Step 6. If NodeSet ! = NULL then
For Each node X in NodeSet do
Construct Schedule [s, Slot[k], X, 1]
Increment k by 1
End For

Step 7. Call Optimal_traversal (Schedule, s)

Step 8. End.

Algortihm 3.0. Tree-Traversals (A, i, j, NodeSet)

//neighbors A contains 1-hop neighbors list of a node
//non_conflict_set [A] contains the list of nodes that donot produce Type 1 or Type 2 interfernces with A

//Possible_frequency_set.—Contains the frequency with which the node A's neighbors can transmit

//ActiveNeighbor [X] contains list of neighbors who are transmitting alogn with X

(1.0) neighbors A[] ← Assign 3-hop, 4-hop. . . n-hop Neighbors list of A

(2.0) non_conflict_set [A] ← Find from neighbors A[] the set of non-conflicting nodes that can transmit along with A.

(3.0) for each node X in non_conflict_set [A] repeat thru steps 4.0 to 8.0.

(4.0) ActiveNeighbor [X] = neighbors [X]

(5.0) Let Possible_ Frequency_set [X] ←
Assign (ActiveNeighbors [X], Frequencyset [])

(6.0) if ActiveNeighbor [X] is having Type 1 or Type 2 interferences with any 1-hop neighbors of each node in non_conflict-set [A] then
AcitveNeighbor [X] = Poweradaption (ActiveNeighbor, X, non_conflict_set, A)

(7.0) Totalnodes [A, j] = Totalnodes [A, j] + number of nodes in ActiveNeighbor [X]

(8.0) End For

(9.0) Return non_conflict_set [A], Totalnodes [A, j]

(10.0) End.

Algorithm 4.0

PowerAdaption (ActiveNeighbor, X, non_conflict_set, A)

//Concurrent$_i$ denotes the list of concurrent nodes transmitting along with node i

Step 1. For each u in ActiveNeighbor $[X]$ do steps 2 thru steps 4

Step 2. AssignLevel = false, conc = concurrent$_u$

Step 3. For each P_s in $P_{\text{low}}, \ldots, P_{\text{high}}$ do
If

$$\left(\frac{(P_s/d_s^{\alpha})}{\left(N^0 + \sum_{j=1}^{\text{conc}} \left(P_r/d_j^{\alpha}\right)\right)} \right) \geq \beta, \tag{5}$$

then

Assign Power P_s to u
Assign Level = true
Break

Step 4. If AssignLevel = false then
ActiveNeighbor $[X]$ = ActiveNeighbor $[X] - u$

Step 5. Return (ActiveNeighbor)

Step 6. End.

Algorithm 5.0 Optimal_traversal (Schedule, s)

Step 1. For I in s repeat steps 2 thru

Step 2. //To Maximize total number of concurrent transmissions
Construct NewSchedule I after Arranging slots in Schedule I in descending order of the totalnumberofnodes for each slot.

Step 3. //To Minimize frame length
Remove a slot consisting of minimum number of nodes from NewSchedule I if all its nodes appear in other remaining slots in NewSchedule I

Step 4. End For

Step 5. DescSort (NewSchedule, s)

Step 6. OptimalSchedule = NewSchedule [1]

Step 7. End.

4. Results

In [11] the authors assumed a 1000 × 1000 square meter rectangular simulating area. The number of nodes in the simulation varied from 2 to 40. They tested with 50 different topologies where each mobile station can tune to its required transmission power. They observed that the average time slots increase linearly as the number of nodes increases for all three algorithms. And they also observed that the Greedy has the smallest schedule length and the RTS/CTS algorithm generates the longest schedule and that the performance difference between these three algorithms is insignificant for small network sizes (less than 17 nodes) and is larger when the network becomes large.

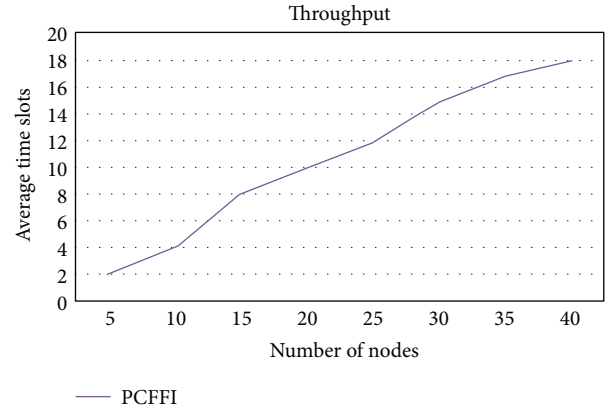


Figure 1: Throughput from PCCFI Simulating area 1000 × 1000 square meter rectangular area, where $N = 5, 10, 15, \ldots, 40$.

In this paper we assumed the same simulating parameters and we obtained the number of time slots from the PCCFI as 4, 8, 10, 12, 15, 17, and 18 when N = 10, 15, 20, 25, 30, 35, and 40, respectively. These results are shown in Figure 1. From [11] we observed that the numbers of time slots of the Greedy, RTS/CTS, and Ranked Schedule methods are far less than 4, 8, 10, 12, 15, 17, and 18 when N = 10, 15, 20, 25, 30, 35, and 40, respectively. Hence these results convey that the PCCFI method outperforms nearly 30–50% of the existing Greedy, Ranked Schedule, and RTS/CTS methods.

In [15] the authors assumed a 200-square area, with 25 m of transmission range. The SDA [16], PAS [17], DAS [18], SAS [19] and First-Fit [20] algorithms are executed with different densities and the results for an average of 10 runs are reported. They compared these algorithms performance with WIRES-BSPT [15]. Since First-Fit algorithm does not produce a conflict free schedule, it is omitted from the evaluations. They concluded that WIRESBSPT [15] outperforms all other solutions by 10% to 30%.

In this paper we assumed the same simulating parameters and with different datasets. The results are shown in Figure 2. These results show that the maximum time slots required by the proposed method are 10, 18, 28, 45, and 65 when N = 20, 40, 60, 80, and 100, respectively.

The maximum time slots required by SDA [16], PAS [17], DAS [18], SAS [19], WIRES-BSPT [15], and First-Fit [20] are less than 10, 18, 28, 45, and 65 when N = 20, 40, 60, 80, and 100, respectively. Hence our proposed PCCFI method outperforms the SDA [16], PAS [17], DAS [18], SAS [19], WIRES-BSPT [15], and First Fit methods by 30–50%.

Hence our proposed algorithm outperforms the results obtained from Greedy, RTS/CTS, Ranked Schedule, SDA, PAS, DAS, SAS, WIRES-BSPT, and First Fit by 30–50%.

5. Conclusion

In this paper we proposed a new network model for the USN called power control collision interference free model. We also presented a distributed data aggregation scheduling protocol using the proposed model which is adaptive to the rate and power for USN.

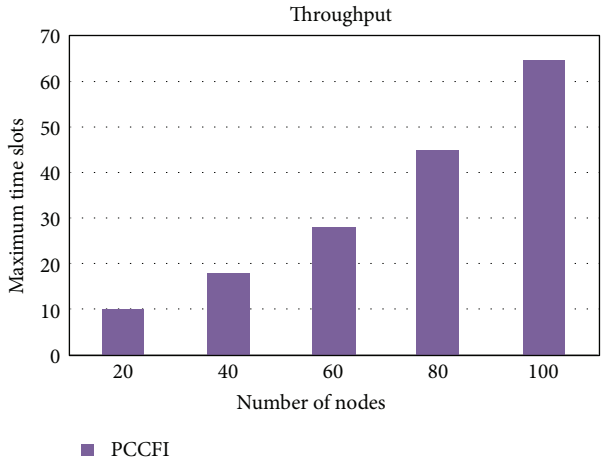


Figure 2: Throughput from PCCFI Simulating area 200 square area, where $N = 20, 40, \ldots, 100$.

Generally there is a high correlation between the transmission power of a node, number of concurrent transmitters and their spatial distances. Larger transmission powers results high data rates but allow a few concurrent transmitters. Similarly smaller transmission powers results low data rates but allow more concurrent transmitters. Therefore, the transmission powers are adjusted according to the number of concurrent transmitters and their distances to yield a maximum network capacity.

References

[1] N. Park, R. Gadh, S. Kim, and D. Won, "System framework and its application in mobile RFID Service network," *Selected Topics in Communication Networks and Distributed Systems* , vol. 113, 2010.

[2] H.-C. Lee, J.-H. Hwang, and H. Yoe, "Energy efficient MAC protocol for ubiquitous agriculture," *International Journal of Smart Home*, vol. 4, no. 3, p. 15, 2010.

[3] W. H. Liao, K. P. Shih, and W. C. Chung, "Multi-channel medium access control protocol with channel distribution for mobile ad hoc networks," *Communications IET*, vol. 3, no. 12, pp. 1821–1831, 2009.

[4] Sharad, S. Mishra, A. K. Sharma, and D. S. Chauhan, "Analysis on energy optimized data collection in tree based ad-hoc sensor network," *International Journal of Modern Education and Computer Science*, pp. 42–49, 2012.

[5] P. J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," *Mobile Networks and Applications*, vol. 9, no. 2, pp. 141–149, 2004.

[6] S. Kumar and S. Chauhan, "A survey on scheduling algorithms for wireless sensor networks," *International Journal of Computer Applications*, vol. 20, no. 5, article 2, 2011.

[7] T. Kim, N. Park, P. K. Chong, J. Sung, and D. Kim, "Distributed low power scheduling in wireless sensor networks," in *Proceedings of the 2nd International Symposium on Wireless Pervasive Computing (ISWPC '07)*, pp. 268–273, San Juan, Puerto Rico, USA, February 2007.

[8] S. Madhavi, "A new adaptive distributed broadcast scheduling method for mobile AdHoc networks," in *Proceedings of the 4th International Conference IEEE Digital Xplore, Networking and Services (ICNS '08)*, pp. 7978–7120, March 2008.

[9] S. Madhavi and I. RameshBabu, "Maximizing the channel throughput in mobile adhoc networks using mIDS," *International Journal of Wireless Communications*, Issue September 2009.

[10] S. Madhavi and I. RameshBabu, "An ACSP interceptor for improving network capacity in mobile adhoc networks," *Masaum Journal of Computing*, vol. 1, no. 3, pp. 382–388, 2009.

[11] H. Choi, J. Wang, and E. A. Hughes, "Scheduling for information gathering on sensor network in Wireless Networks," *Wireless Networks*, vol. 15, no. 1, pp. 127–140, 2009.

[12] R. Sheikhpour, S. Jabbehdari, and A. K. Zadeh, "A cluster-chain based routing protocol for balancing energy consumption in wireless sensor networks," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 7, no. 2, 2012.

[13] A. K. Tripathy and S. Chinara, "Comparison of residual energy-based clustering algorithms for wireless sensor network," *ISRN Sensor Networks*, vol. 2012, Article ID 375026, 10 pages, 2012.

[14] T. Kim, H. Lim, and J. C. Hou, "Improving spatial reuse through tuning transmit power, carrier sense threshold and data rate in multihop wireless networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking (MobiCom '06)*, pp. 366–377, ACM, Los Angeles, Calif, USA, September 2006.

[15] B. Malhotra, I. Nikolaidis, and M. A. Nascimento, "Aggregation convergecast scheduling in wireless sensor networks," *Wireless Networks*, vol. 17, no. 2, pp. 319–335, 2011.

[16] X. Chen, X. Hu, and J. Zhu, "Minimum data aggregation time problem in wireless sensor networks," *Lecture Notes in Computer Sciences*, vol. 3794, pp. 133–142, 2005.

[17] X. Yu, S. Mehrotra, and N. Venkatasubramanian, "Sensor scheduling for aggregate monitoring in wireless sensor networks," in *Proceedings of the 19th International Conference on Scientific and Statistical Database Management (SSDBM '07)*, p. 24, 2007.

[18] B. Yu, J. Li, and Y. Li, "Distributed data aggregation scheduling in wireless sensor networks," in *Proceedings of the 28th Conferenceon Computer Communications (INFOCOM '09)*, pp. 2159–2167, Rio de Janeiro, Brazil.

[19] P. J. Wan, S. C. H. Huang, L. Wang, Z. Wan, and X. Jia, "Minimum latency aggregation scheduling in multihop wireless networks," in *Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '09)*, pp. 185–194.

[20] S. C. H. Huang, P. J. Wan, C. T. Vu, Y. Li, and F. Yao, "Nearly constant approximation for data aggregation scheduling in wireless sensor networks," in *Proceedings of the 26th Conference on Computer Communications (INFOCOM '07)*, pp. 366–372.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 149023, 22 pages
http://dx.doi.org/10.1155/2013/149023

Review Article

Detection and Mitigation of Node Replication Attacks in Wireless Sensor Networks: A Survey

Wazir Zada Khan,[1] Mohammed Y. Aalsalem,[2] Mohammed Naufal Bin Mohammed Saad,[1] and Yang Xiang[3]

[1] *Electrical and Electronic Engineering Department, Universiti Teknologi PETRONAS, Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia*
[2] *School of Computer Science & Information System, Jazan University, Jazan 45142, Saudi Arabia*
[3] *School of Information Technology, Deakin University, 221 Burwood Highway, Burwood, Melbourne, VIC 3125, Australia*

Correspondence should be addressed to Wazir Zada Khan; wazirzadakhan@yahoo.com

Received 25 January 2013; Accepted 22 March 2013

Academic Editor: Muhammad Khurram Khan

Wireless sensor networks are a collection of a number of tiny, low-cost, and resource-constrained sensor nodes which are commonly not tamper proof. As a result, wireless sensor networks (WSNs) are prone to a wide variety of physical attacks. In this paper, we deem a typical threat known as node replication attack or clone node attack, where an adversary creates its own low-cost sensor nodes called clone nodes and misinforms the network to acknowledge them as legitimate nodes. To instigate this attack, an adversary only needs to physically capture one node, and after collecting all secret credentials (ID, cryptographic keys, etc.), an adversary replicates the sensor node and deploys one or more clones of the compromised node into the network at strategic positions, damaging the whole network by carrying out many internal attacks. Detecting the node replication attack has become an imperative research topic in sensor network security, and designing detection schemes against node replication attack involves different threatening issues and challenges. In this survey, we have classified the existing detection schemes and comprehensively explore various proposals in each category. We will also take a glance at some technical details and comparisons so as to demonstrate limitations of the existent detections as well as effective contributions.

1. Introduction

Advancement in technology has made it possible to develop tiny low-cost sensor nodes with off-the-shelf hardware. A wireless sensor network (WSN), which is a distributed and self-organized network, is a collection of such sensor nodes with limited resources that collaborate in order to achieve a common goal. These sensor nodes are comprised of low-cost hardware components with constraints on battery life, memory size, and computation capabilities [1]. Wireless sensor networks are often deployed in harsh and hostile environments which are inaccessible and even hazardous areas to perform various monitoring tasks. For example, they can be used to monitor factory instrumentation, pollution levels, freeway traffic, and the structural integrity of buildings [2]. Some of the other applications of WSNs include patient monitoring, climate sensing, control in office buildings, and home environmental sensing systems for temperature light, moisture, and motion.

WSNs are viable solutions for a wide variety of real-world challenges; however, a set of new security challenges arise in sensor networks due to the fact that current sensor nodes lack hardware support for tamper-resistance (because it is uneconomical to enclose each node in a tamper resistant hardware) and are often deployed in unattended environments where they are vulnerable to capture and compromise by an adversary. Taking an example of a battlefield, WSNs must tackle the threats and attacks from attackers because these areas are sometimes physically accessible to camouflaged enemies [3] who would like to acquire the private locations of soldiers from or inject wrong commands into the sensor network [4]. Similarly, an unattended WSN can be deployed

in hostile environments which imply the existence of an adversary. For example, WSN can be used to monitor firearm discharge, illicit crop cultivation, drug/weapons smuggling, human trafficking, nuclear emissions in a rogue region and other illegal activities [5]. Thus, it is very important to ensure the security of sensor networks in such scenarios.

The unattended nature of wireless sensor networks can be exploited by adversaries which are able to launch an array of different physical attacks including node replication attack, signal or radio jamming, denial of service (DoS) attack, node outage, eavesdropping, and Sybil attack. and other attacks like sinkhole, wormhole, and selective forwarding attack. Threats to sensor networks can be either layer dependent or layer independent. Attacks in the former category can be application dependant and are specific to different OSI layers targeting specific network functionalities such as routing, node localization, time synchronization, and data aggregation, while the attacks in the latter category are application independent affecting a wide variety of applications from object tracking and fire alarming to battlefield surveillance, and these attacks are not launched on any OSI layer. The attacks of the latter category are also application independent [2]. This attack taxonomy is also shown in Figure 1. In order to protect wireless sensor networks from layer dependent attacks, many schemes have been proposed. To alleviate the effects of routing disruption attacks, secure routing schemes have been proposed [6, 7]. Authentication schemes [8–10] are used to mitigate false data injection attacks. Data aggregation can be secured by using secure data aggregation protocols proposed in [11–14]. To defend localization and time synchronization protocols from different attacks, and threats many protocols have been proposed in [15–21]. Nevertheless, most of these schemes are attack resilient, rather than they can detect and remove the source of attack. Thus, there is a need to detect and revoke the sources of attacks as soon as possible to substantially reduce the costs and damages incurred by employing attack resilient approaches.

In this comprehensive survey, we consider a very severe and important physical attack on WSN which is called node replication attack or clone attack. It is also known as identity attack. In this attack, an adversary first physically captures only one or few of legitimate nodes, then clones or replicates them fabricating those replicas having the same identity (ID) with the captured node, and finally deploys a capricious number of clones throughout the network. This whole process of node replication attack and the various stages are shown in Figure 2. This vexing problem arises from the actuality that sensor nodes are unshielded. It is stated in [22] that an experienced attacker can completely compromise a typical sensor node by using only a few readily available tools, and it can then obtain copies of that node memory and data within 1 min of discovering it. The clones or replicas may even be selectively reprogrammed to subvert the network by launching further insider attacks like falsifying sensor data or suppressing legitimate data, extracting data from the network and disconnect the network by triggering correct execution of node revocation protocols that rely on threshold voting schemes and staging denial of service (DoS) attacks. Clone nodes may create a black hole, initiate a wormhole attack

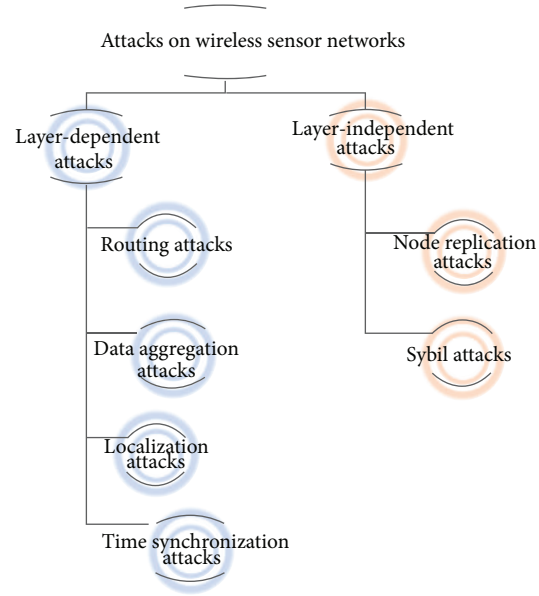


Figure 1: Classification of attacks on wireless sensor networks.

with a collaborating adversary, or may also leak data in an environment in which sensed data must be kept private [23]. If these replicated nodes or clones remain undetected or unattended for a long time, they can further commence the changes in protocol behavior and intrusion into the systems security [24]. It is easy for an adversary to launch such attacks due to the fact that the clones, created by an adversary, have legitimate information (codes, key materials, and credentials), and they may be considered as legitimate nodes and totally honest by its neighbors which are participating in the network operation in the same way as the noncompromised nodes.

The above mentioned traditional security schemes for WSNs are inept to detect and prevent node replication attack. Thus, in the last few years, a number of detection and prevention techniques/schemes have been proposed in the literature. According to [2], the detection schemes are classified on a high level as network-based or radio-based detection. Only one instance of radio-based detection is found in [25]. The former category is further categorized into two types as for mobile WSNs and for stationary WSNs. Both techniques for mobile and stationary WSNs are further divided into two broad categories, namely, centralized and distributed. This can be summarized with Figure 3 which shows a detailed classification of all replica detection schemes. This categorization provides a first step to better understand the node replication detection schemes.

A WSN can be either stationary or mobile. In static wireless sensor networks (SWSNs), the sensor nodes are stationary or static; that is, the sensor nodes are deployed randomly, and after deployment their positions do not change. On the other hand, in mobile wireless sensor networks (MWSNs), the sensor nodes can move on their own, and after deployment, they can interact with the physical environment by controlling their own movement. Advances in robotics have made it possible to develop such mobile sensors which

Figure 2: Steps of node replication attack.

are autonomous and have the ability to sense, compute, and communicate like static sensors. The prime difference between static and mobile WSNs is that mobile nodes are able to reposition and organize themselves in the network, and after initial deployment, the nodes spread out to gather information [26, 27]. Mobile nodes can communicate with one another when they are within the range of each other, and only then they can exchange their information gathered by them. Another important difference is that in static WSNs fixed routing or flooding is used for data distribution, while in mobile WSNs dynamic routing is used. As static and mobile WSNs differ in their characteristics hence replication detection schemes for stationary and mobile WSNs will be substantially different. In a static or stationary WSN, a sensor node has a unique deployment position, and thus if one logical node ID is found to be associated with two or more physical locations, node replication is detected. But this is inapplicable to mobile WSNs where sensor nodes keep roaming in the deployment field. So, replication detection in such mobile WSN involves different scenarios and techniques.

For mobile WSNs, both centralized and distributed techniques have been proposed in the literature. In the case of stationary WSNs, centralized techniques are further categorized into five types, namely, straightforward base station-based technique, key usage-based technique, SET operations techniques, cluster head-based techniques and neighborhood social signature-based techniques. The distributed techniques for stationary WSNs are further divided into four types naming Node to Network Broadcasting, claimer-reporter-witness-based techniques, neighbor-based and generation- or group-based techniques. On the other hand, mobile centralized detection techniques are further divided into two types including key usage-based and node speed-based techniques. The mobile distributed detection techniques are divided into three main types, namely, node meeting-based, mobility-assisted-based, and information-exchange-based techniques. This inclusive categorization can be summarized with Figure 3 which provides a first step in better understanding node replication detection schemes.

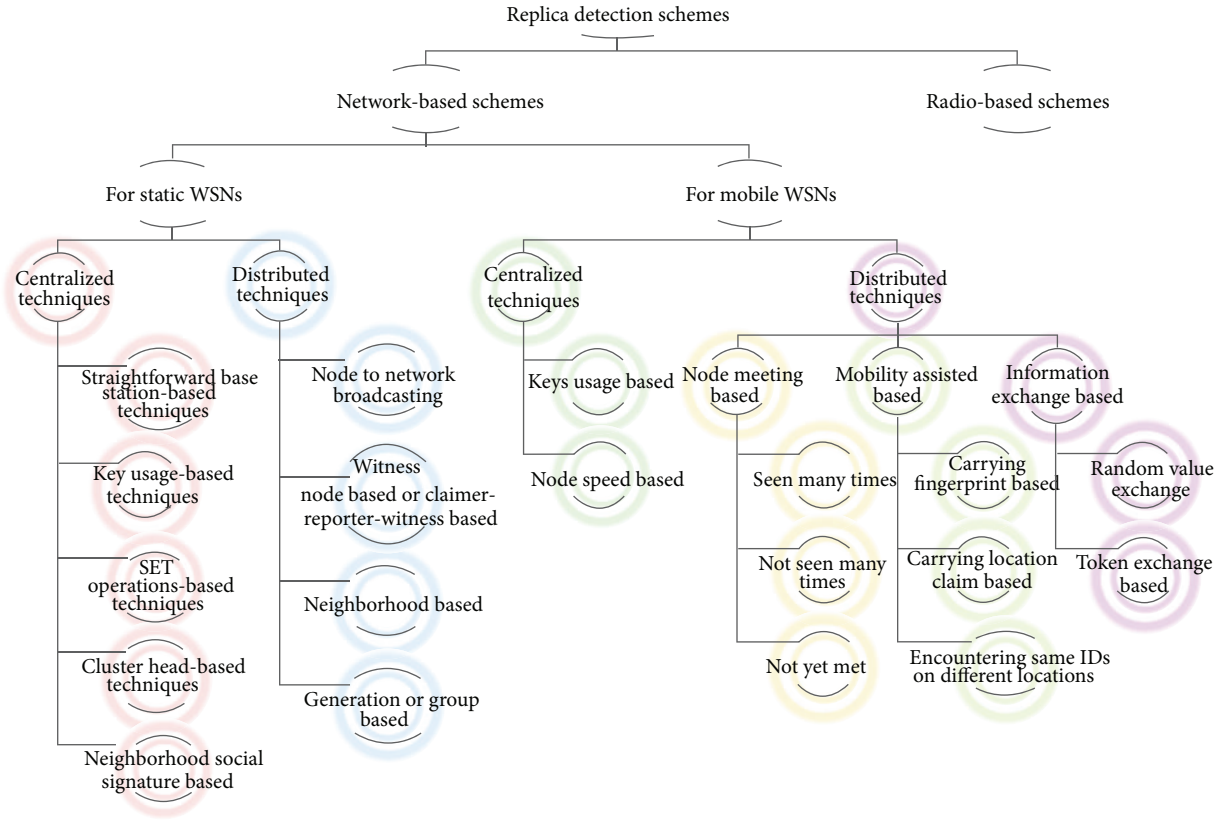


Figure 3: Taxonomy of replica detection schemes.

1.1. Motivation. With the rapid use of vast technologies in WSNs, the threats and attacks to WSN are escalating and are also being diversified and deliberate. A typical threat called node replication attack is a very severe and niggling problem in which an adversary replicates a sensor node after physically capturing it and then uses these replicas to disrupt the network operations by redeploying them at strategic positions of the network. Thus the research related to node replication attack in WSNs has been followed with much interest in recent years. The research of authentication and security techniques is already quite mature but such solutions fail to detect node replication attack and thus no longer provide WSN with adequate security from this attack. Furthermore, the detection of node replication attack in mobile WSN is far different and more challenging than in static WSNs.

The development of replica/clone detection techniques suitable for static WSNs and mobile WSNs is therefore regarded as an essential research area which will make WSN (either static or mobile) to be more secure and reliable. Most recently, Zhu et al. [2] did a survey on the countermeasures of node replication attack which has pointed out some valuable technical weaknesses and advantages of some of the techniques, but latest progress of replica detection schemes is absent, and it also lacks the detailed analysis of all existing techniques for mobile WSNs.

This motivates us to present our paper as a complete guideline of replica/clone detection schemes both for static and mobile WSNs. Moreover, in this paper we have identified the advantages and shortcomings of all the techniques/schemes. Finally, some variations of node replication attack are also identified and discussed. This paper is helpful in understanding all the replica detection schemes developed so far, and it can assist the researchers and developers in the development of new, robust, and effective detection schemes.

1.2. General Adversary Model. Conventionally, some assumptions are made about an adversary in order to scrutinize security of a sensor network. First of all, an adversary is a smart and powerful attacker who can launch a clone attack [4], and it has the ability to secretly capture a limited number of legitimate sensor nodes [28]. Secondly, an adversary can create replicas by using cryptographic information which is obtained from the compromised node. An adversary has also full control over the compromised and replicated nodes and can communicate with them at any time. Thirdly, the main goal of an adversary is to protect its replicas from being detected by the detection protocol used in the network because if any replicas are detected, besides starting a revoke process to revoke replicas, the network may start a sweeping process to sweep out [29] the compromised node and may also draw human intervention. Thus, it is mostly assumed that nodes controlled by an adversary still follow the replica detection protocol as an adversary always wants to be overlooked. Fourthly, an adversary is so powerful that it is able to subvert

the nodes that will possibly act as witnesses. To cope with such an adversary, it could be possible to assume that nodes are tamper-proof. But as tamper proof hardware is expensive and energy demanding, a large part of the literature has assumed that nodes in the network are not tamper resistant.

In case of mobile WSNs, the method of attack is the same but difference is that an adversary is mobile. The scenario of mobile WSN is that the sensors are unable to transmit sensed data at their will because the sink is not always present. Thus, the data accumulated in their memories become targets of many adversaries. In [30], a mobile adversary model is proposed in which mobile adversary visits and travels around the network trying to compromise a subset of sensors within the time interval when sinks are not present in the network. The time taken by a mobile adversary to compromise a set of sensors is much shorter than the time between two successive data collections of a sink.

1.3. Node Replication Attack and Its Effects on the Security Goals of WSNs. High level security issues are basically identical to the security requirements of both static and mobile WSNs. Thus, when dealing with security of WSNs, one is faced with achieving some of the following common security goals including availability, authenticity, confidentiality, and data integrity. When node replication attack is launched by an adversary, all of these security goals are affected severely because of two reasons. First, if any proper, specific, and efficient detection scheme is not used to identify and revoke these replicas because the existing general purpose security protocols would allow the replica nodes to encrypt, decrypt, and authenticate all of their communications as if they were original captured nodes. Second, when the detection probability of the detection technique used is very low to detect these clones or replicas. Node replication attack is significantly harmful to the networks because these replicas or clones have legitimate keys, and they are recognized as legitimate members of the network, since they carry all cryptographic materials extracted from the captured nodes so that an adversary can use them to mount a variety of insider attacks [2]; for example, it can monitor all the information passing through the nodes or monitor significant fraction of the network traffic that passes through the nodes, falsify sensor data, launch denial of service (DoS) attack, extract data from the network, inject false data to corrupt the sensor's monitoring operation, subvert data aggregation, and jam legitimate signals and can also cause continual disruption to network operations by undermining common network protocols.

Availability ensures the survivability of network services despite attacks [31]. In case of node replication attack, an adversary is able to compromise the availability of WSN by launching a denial of service (DoS) attack, which can severely hinder the network's ability to continue its processing. By jamming legitimate signals, the availability of the network assets to authorized parties is also affected.

Authenticity is a security goal that enables a node to ensure the identity of the sensor node it is communicating with. In case of node replication attack, an adversary creates clone nodes which are seemingly legitimate ones (identical to the original captured node) as they have all the secret credentials of the captured node; thus, it is difficult for any node to differentiate between a clone node and the original or legitimate node. Also the existing authentication techniques cannot detect clone nodes as they all hold legitimate keys. This is how the authenticity of the network is affected.

Confidentiality is the assurance that sensitive data is being accessed and viewed only by those who are authorized to see it. But when node replication attack is launched, confidentiality of data is not assured as clone nodes are the duplicated nodes of the compromised ones, and thus they behave like original compromised nodes. These clone nodes can have all the data that contains trade secrets for commercial business, secret classified government information, or private medical or financial records, and thus by misusing such sensitive data, it can damage the network or organization, person, and governmental body.

Data integrity ensures that the contents of data or correspondences are preserved and remain unharmed during the transmission from sender to receiver. Integrity represents that there is a guarantee that a message sent is the message received meaning that it was not altered either intentionally or unintentionally during transmission. But in case of node replication attack, an adversary can falsify sensor data or can inject false data to corrupt the sensitive data and thus subverting the data aggregation using the replicated or clone nodes.

1.4. Evaluation Metrics for Replication Detection Techniques. For the performance analysis and evaluation of replica detection protocols, four vital evaluation metrics are mostly used by all the detection schemes. These are communication overhead, storage or memory overhead, detection probability and detection time [26].

Communication overhead is defined as the average number of messages sent by a sensor node while propagating the location claims. *Storage overhead* defines the average number of the location claims stored in a sensor node. *Detection probability* is an important evaluation metric which shows how accurately a protocol can identify and detect the clones or replicas. The *detection time* is simply the delay between actual replica node deployment and detection.

To make the current survey more comprehensive and detailed, here in Section 2 we have discussed all the existing schemes for the replica detection in stationary WSNs which are accordingly compared in Section 3. Section 4 describes all the replication detection schemes in mobile WSNs proposed so far in the literature which are then compared in Section 5. In Section 6, we have highlighted some important issues and challenges associated with the node replication attack in both static and mobile WSNs. Finally, Section 7 concludes the paper.

2. Detection Techniques for Stationary WSNs

Many techniques have been proposed for the detection of node replication attack in static WSNs which are categorized mainly into two types as centralized and distributed techniques.

2.1. Centralized Techniques. In centralized techniques base station is considered to be a powerful central which is responsible for information convergence and decision making. During the detection process every node in the network sends its location claim (ID, Location Info) to base station (sink node) through its neighboring nodes. Upon receiving the entire location claims, the base station checks the node IDs along their location, and if it finds two different locations with the same ID, it raises a clone node alarm.

2.1.1. On the Detection of Clones in Sensor Networks Using Random Key Predistribution. This technique falls into the category of key usage based techniques. Brooks et al. [32] have proposed a cloned key detection protocol in the context of random key predistribution [33]. The basic idea is that the keys employed according to the random key predistribution scheme should follow a certain pattern, and those keys whose usage exceeds a threshold can be judged to be cloned. In the protocol, counting Bloom filters is used to collect key usage statistics. Each node makes a counting Bloom filter of the keys it uses to communicate with neighboring nodes. It appends a random number (nonce) to the Bloom filter and encrypts the result using base station public key; this encrypted data structure is forwarded to base station. Base station decrypts the Bloom filters it receives, discards duplicates, and counts the number of time each key used in the network. Keys used above a threshold value are considered cloned. Base station makes a bloom filter from the cloned keys, encrypts the list using its secret key and broadcasts this filter to the sensor network using a gossip protocol. Each node decrypts base stations bloom filter removes cloned keys from its keying, and terminates connections using cloned keys.

2.1.2. SET: Detecting Node Clones in Sensor Networks. This technique falls into the category of base station-based techniques. Choi et al. [23] have proposed a clone detection approach in sensor networks called SET. In SET, the network is randomly divided into exclusive subsets. Each of the subsets has a subset leader, and members are one hop away from their subset leader. Multiple roots are randomly decided to construct multiple subtrees, and each subset is a node of the subtree. Each subset leader collects member information and forwards it to the root of the subtree. The intersection operation is performed on each root of the subtree to detect replicated nodes. If the intersection of all subsets of a subtree is empty, there are no clone nodes in this subtree. In the final stage, each root forwards its report to the base station (BS). The BS detects the clone nodes by computing the intersection of any two received subtrees. SET detects clone nodes by sending node information to the BS from subset leader to the root node of a randomly constructed subtree and then to the BS.

2.1.3. Real-Time Detection of Clone Attacks in Wireless Sensor Networks. This technique falls into the category of neighborhood social signature-based techniques. Xing et al. [34] have proposed real-time detection of clone attacks in WSN. In their approach, each sensor computes a fingerprint by incorporating the neighborhood information through a superimposed s-disjunct code [35]. Each node stores the fingerprint of all neighbors. Whenever a node sends a message, the fingerprint should be included in the message, and thus neighbors can verify the fingerprint. The messages sent by clone nodes deployed in other locations will be detected and dropped since the fingerprint does not belong to the same "community." The motivation behind their scheme for detection of clone attacks is exploring the social characteristics of each sensor. Once they are deployed, these sensors reside within a fixed neighborhood. The sensor and its neighborhood form a small "community," or a "social network." A cloned sensor can have the same legitimate credentials (ID, keys, etc.) as the original node, but cannot have the same community neighborhood. Thus, each sensor can be distinguishably characterized by its social community network. In a small community, a newcomer can be easily recognized if speaking with a different accent. Similarly, a clone node can be easily identified by its neighbors if carrying a "social signature" belonging to a different community.

2.1.4. Hierarchical Node Replication Attacks Detection in Wireless Sensor Networks. This technique falls into the category of cluster head-based techniques. Znaidi et al. [36] have proposed a cluster head selection-based hierarchical distributed algorithm for detecting node replication attacks using a Bloom filter mechanism including the network reactions. More precisely, the algorithm relies on a cluster head selection performed using the local negotiated clustering algorithm (LNCA) protocol [37]. Each cluster head exchanges the member node Ids through a Bloom filter with the other cluster heads to detect eventual node replications. The algorithm works in three steps. In the first step all the material required for Bloom filter computations and for cryptographic operations that will be performed in the network predistributed in each sensor node. The second step performs the cluster head election. In the third step, Bloom filter construction is performed by each cluster head, and the Bloom filter verification is performed by the other cluster heads.

2.1.5. CSI: Compressed Sensing-Based Clone Identification in Sensor Networks. This technique falls into the category of base station-based techniques. Yu et al. [38] have proposed a centralized technique called compressed sensing-based clone identification (CSI) for static wireless sensor networks. The basic idea behind CSI is that each node broadcasts a fixed sensed data (α) to its one hop neighbors. Sensor nodes forward and aggregate the received numbers from descendant nodes along the aggregation tree via compressed sensing-based data gathering techniques. Base station (BS), as the root of the aggregation tree, receives the aggregated result and recovers the sensed data of the network. According to the reconstructed result, the node with the sensory reading greater than α is the clone since a nonclone node can only report the number once.

2.2. Distributed Techniques. In distributed techniques, no central authority exists, and special detection mechanism called claimer-reporter-witness is provided in which the detection is performed by locally distributed node sending

the location claim not to the base station (sink) but to a randomly selected node called witness node. Distributed techniques are classified into four types and these are described below.

2.2.1. Node-to-Network Broadcasting (N2NB) and Deterministic Multicast (DM). This technique falls into the category of node-to-network broadcasting. The N2NB and DM protocols are two unappealing examples proposed by Parno et al. [28]. Both of protocols received relatively less attention. In N2NB, each node floods the entire network with authenticated broadcast to claim its own location (instead of its neighbors). Each node stores the location information for its neighbors, incurring a storage cost of $O(d)$. Each node upon receiving a conflicting claim invokes a revocation procedure against the offending nodes, and eventually any replica will be cut off by all its neighbors (thus isolated from the WSN). The N2NB protocol achieves 100% detection rate as long as the broadcast reaches every node if the network size is assumed to be n and certain duplicate suppression algorithm is employed so that each node only broadcasts a given message once.

The DM protocol is a good example to illustrate the claimer-reporter-witness framework. The claimer is a node which locally broadcasts its location claim to its neighbors, each neighbor serving as a reporter, and employs a function to map the claimer ID to a witness. Then the neighbor forwards the claim to the witness, which will receive two different location claims for the same node ID if the adversary has replicated a node. One problem can occur that the adversary can also employ the function to know about the witness for a given claimer ID, and may locate and compromise the witness node before the adversary inserts the replicas into the WSN so as to evade the detection.

2.2.2. Distributed Detection of Node Replication Attacks in Sensor Networks. Both RM and LSM fall into the category of witness node-based techniques. Parno et al. [28] have introduced two more distributed algorithms for the detection of clone nodes in wireless sensor networks which are quite mature schemes as compared to DM. The first protocol is called randomized multicast (RM) which distributes location claims to a randomly selected set of witness nodes. The birthday paradox [39] predicts that a collision will occur with high probability if the adversary attempts to replicate a node. Their second protocol, line-selected multicast (LSM), exploits the routing topology of the network to select witnesses for a node location and utilizes geometric probability to detect replicated nodes.

In RM, each node broadcasts a location claim to its one-hop neighbors. Then, each neighbor selects randomly witness nodes within its communication range and forwards the location claim with a probability to the nodes closest to chosen locations by using geographic routing. At least one witness node is likely to receive conflicting location claims according to birthday paradox when replicated nodes exist in the network. In LSM, the main objective is to reduce the communication costs and increase the probability of detection. Besides storing location claims in randomly selected witness nodes, the intermediate nodes for forwarding location claims can also be witness nodes. This seems like randomly drawing a line across the network, and the intersection of two lines becomes the evidence node of receiving conflicting location claims.

2.2.3. A New Protocol for Securing Wireless Sensor Networks against Node Replication Attacks. This technique falls into the category of generation- or group-based techniques. Bekara and Laurent-Maknavicius [40, 41] have proposed a new protocol for securing WSN against node replication attack by limiting the order of deployment using symmetric polynomial for pair-wise key establishment and defined group-based deployment model. Their scheme requires sensors to be deployed progressively in successive generations (or group). Each node belongs to a unique generation. In their scheme, only newly deployed nodes are able to establish pairwise keys with their neighbors, and all nodes in the network know the number of the highest deployed generation. Therefore, the clone nodes will fail to establish pair-wise keys with their neighbors since the clone nodes belong to an old deployed generation.

2.2.4. A Randomized, Efficient, and Distributed Protocol for the Detection of Node Replication Attacks in Wireless Sensor Networks. This technique falls into the category of witness node-based techniques. Conti et al. have proposed a randomized, efficient, and distributed protocol called RED [42, 43] for the detection of node replication attack. It is executed at fixed intervals of time and consists in two steps. In first step, a random value, *rand*, is shared between all the nodes through base station. The second step is called detection phase. In the detection phase, each node broadcasts its claim (ID and location) to its neighboring nodes. Each neighbor node that hears a claim sends (with probability p) this claim to a set of g pseudorandomly selected network locations. The pseudo random function takes as an input ID, random number, and g. Every node in the path (from claiming node to the witness destination) forwards the message to its neighbor nearest to the destination. Hence, the replicated nodes will be detected in each detection phase. When next time the RED executes, the witness nodes will be different since the random value which is broadcasted by the BS is changed.

2.2.5. Efficient Distributed Detection of Node Replication Attacks in Sensor Networks. These techniques falls into the category of witness node-based techniques. Zhu et al. [44, 45] have proposed two distributed protocols for detecting node replication attacks called single deterministic cell (SDC) and parallel multiple probabilistic cells (P-MPC). In both protocols, the whole sensor network is divided into cells to form a geographic grid. In SDC, each node ID is uniquely mapped to one of the cells in the grid. When executing detection procedure, each node broadcasts a location claim to its neighbors. Then, each neighbor forwards the location claim with a probability to a unique cell by executing a geographic hash function [46] with the input of node ID. Once any node in the destination cell receives the location claim, it floods the location claim to the entire cell. Each node in the

destination cell stores the location claim with a probability. Therefore, the clone nodes will be detected with a certain probability since the location claims of clone nodes will be forwarded to the same cell. Like SDC, in the P-MPC scheme, a geographic hash function [46] is employed to map node identity to the destination cells. However, instead of mapping to single deterministic cell, in P-MPC the location claim is mapped and forwarded to multiple deterministic cells with various probabilities. The rest of the procedure is similar to SDC.

2.2.6. (Space-Time)-Related Pairwise Key Predistribution Scheme for Wireless Sensor Networks. This technique falls into the category of base station-based techniques. Fei et al. [47] have proposed a polynomial based space-time-related pairwise key predistribution scheme (PSPP-PKPS, for short PSPP) for wireless sensor networks, which relates the keying material of a node with its deployment time and location. In PSPP, the keying material of a node can only work at its initial deployment location. If a node leaves its deployment location, its keying material will become invalid. By using this idea, their scheme provides resistance against the clone attack.

2.2.7. A Neighbor-Based Detection Scheme for Wireless Sensor Networks against Node Replication Attacks. This technique falls into the category of neighborhood-based techniques. Ko et al. [48] have proposed a real time neighbor-based detection scheme (NBDS) for node replication attack in wireless sensor networks. The main idea of their scheme is that when a person moves to another community, he will meet new neighbors and tell his new neighbors where he comes from through chatting. But new neighbors will not check if he lies or not. However, if some of his new neighbors ask his previous neighbors whether this newcomer really comes from the community that he claims, the identity of the newcomer can be implicitly verified. If previous neighbors say that this person still lives in the original neighborhood, the newcomer can be detected as a replica. This observation motivates their research on node replication attacks, and replicas are detected in the same way.

2.2.8. Distributed Detection of Node Capture Attacks in Wireless Sensor Networks. This technique falls into the category of base station-based techniques. Ho [49] has proposed a node capture detection scheme for wireless sensor networks. Their scheme detects the captured sensor nodes by using the sequential analysis. They use the fact that the physically captured nodes are not present in the network during the period from the captured time to the redeployment time. Accordingly, captured nodes would not participate in any network operations during that period. By leveraging this intuition, the captured nodes can be detected by using the sequential probability ratio test (SPRT) [50]. The protocol first measures the absence time period of a sensor node and then compares it to a predefined threshold. If it is more than threshold value, the sensor node is considered as a captured node. The efficient node capture detection capability depends on a properly configured threshold value.

2.2.9. Memory Efficient Protocols for Detecting Node Replication Attacks in Wireless Sensor Networks. These techniques fall into the category of witness node-based techniques. Zhang et al. [3] have proposed four memory efficient multicast protocols for replication detection, namely, memory efficient multicast with Bloom filters (B-MEM), memory efficient multicast with Bloom filters and Cell Forwarding (BC-MEM), memory efficient multicast with cross forwarding (C-MEM), and memory efficient multicast with cross and cell forwarding (CC-MEM). The first protocol B-MEM use Bloom filters to compress the information stored at the sensors and the location claim Cα of a node α is multicast via its neighbors to a number of randomly selected locations in the network. Each neighbor β has a probability p to participate in the multicast. If it does, it becomes a witness node and sends Cα to a random location in the network. The node closest to that location will be another witness node w to store Cα. The watcher nodes on the routing path P from β to w only store the membership of IDα and lα in the Bloom filters. Such membership information can help them detect any conflicting location claim C'α received later, and guide C'α along P to either β or w, which will then broadcast both Cα and C'α to the entire network in order to revoke node α and its replicas.

The second protocol BC-MEM is designed on top of B-MEM. It adopts a cell forwarding technique that not only solves the crossover problem but also reduces the memory overhead. The deployment area is divided into virtual cells. In each cell an anchor point is assigned for every node in the network. The anchor point for a node α is determined by α ID. The node closest to the anchor point is called the anchor node for α. In B-MEM, when a location claim is forwarded on a line segment, all intermediate nodes on the line serve as watchers, while the first node and the last node serve as witnesses. In contrast, in BC-MEM a claim is not forwarded on the line segment. It is forwarded to the anchor point in the next cell where the line segment intersects. The claim is forwarded from one anchor node to another until reaching the last cell. The anchor nodes in the intermediate cells are watchers, and the anchor nodes in the first and last cells are witnesses.

The third protocol C-MEM is designed on top of B-MEM. It incorporates a new cross forwarding technique to solve the crowded center problem. B-MEM stores the information about a location claim along randomly selected line segments, which are likely to pass the center area of the deployment. On the other hand, C-MEM first selects a random point (called the cross point) in the network and forwards the location claim to that point. From there, it forwards the claim along the horizontal and vertical lines that pass the cross point. While the node closest to the cross point is a witness node, the nodes along the horizontal and vertical lines are watchers. Since the cross points for all location claims are distributed uniformly at random in the network, it is no longer true that the lines pass the center area more frequently. C-MEM does not use cell forwarding.

The fourth protocol CC-MEM combines cross forwarding and cell forwarding to solve both the crowded center problem and the crossover problem, such that it can detect

node replication attack with high probability and low overhead.

2.2.10. Active Detection of Node Replication Attacks. This technique falls into the category of base station-based techniques. Melchor et al. [51] have proposed a distributed protocol for the detection of replication attack for wireless sensor networks, in which each node verifies at random a few other nodes in the network. The proposed protocol does not build a distributed database of location claims that will contain local conflicting claims when replicas exist. The idea is that each node will actively test if 1 k other random nodes are replicated or not; they call them the scrutinized nodes. In order to test whether a scrutinized node α is replicated or not, 2 k nodes are randomly chosen in the network and asked to forward to α a request for a signed location claim. If two replicas exist, each will probably receive a request, and if both answer, two conflicting claims will be obtained by the queerer.

2.2.11. Randomly Directed Exploration: An Efficient Node Clone Detection Protocol in Wireless Sensor Networks (RDE). This technique falls into the category of witness node-based techniques. Zhijun et al. [52] have presented a novel clone node detection protocol called randomly directed exploration. This protocol does not call for any unrealistic assumptions. Each node only needs to know its neighbor nodes. During the detection procedure, nodes issue claiming messages containing neighbor list with a maximum hop limit to randomly selected neighbors. The previous transmission of a claiming message forms a direction, and then the intermediate node tries to follow the direction to forward the message. During forwarding messages, the intermediate nodes explore the claiming messages for node clone detection. In such a simple way, the proposed protocol can efficiently detect clone nodes in the dense sensor networks. In addition, the protocol consumes almost minimum memory during detection, and communication payload is satisfactory. It can scale to large configurations. They have implemented the protocol in the OMNet++ simulation framework.

2.2.12. Random-Walk-Based Approach to Detect Clone Attacks in Wireless Sensor Networks. These two techniques fall into the category of witness node-based techniques. Zeng et al. [4] have proposed two protocols RAndom WaLk (RAWL) and Table-assisted RAndom WaLk (TRAWL) for the detection of clone attack in wireless sensor networks. The RAndom WaLk (RAWL) starts several random walks randomly in the network for each node a, and then selects the passed nodes as the witness nodes of node a. RAWL works in four steps in each execution. In the first step, each node broadcasts a signed location claim. In the second step, each of the node neighbors probabilistically forwards the claim to some randomly selected nodes. In the thirds step, each randomly selected node sends a message containing the claim to start a random walk in the network, and the passed nodes are selected as witness nodes and will store the claim. In the fourth step, if any witness receives different location claims for same node ID, it can use these claims to revoke the replicated node.

The second protocol, Table-assisted RAndom WaLk (TRAWL), is based on RAWL and adds a trace table at each node to reduce memory cost. Usually, the memory cost is due to the storage of location claims, but in TRAWL each node only stores $O(1)$ location claims (although the size of the trace table is still $O(\sqrt{n}\log n)$, the size of a table entry is much smaller than the size of a location claim). When a randomly chosen node starts a random walk, all the passed nodes will still become witness nodes. However, now they do not definitely store the location claim, instead, they store the location claim independently with probability $c_2\sqrt{n}\log n$, where c_2 is a constant. Also, each witness node will create a new entry in its trace table for recording the pass of a location claim.

2.2.13. CINORA: Cell-Based Identification of Node Replication Attack in Wireless Sensor Networks. Gautam Thakur [24] has proposed two distributed methods for detecting node replication attack based on intersecting sets called CINORA-Inset and restricted cell two-phase authentication model called CINORA-Hybrid. Initially, the sensor network is divided into geographical cells similar to the existing cellular network. However, their approach does not deterministically map a nodes identity to a cell. In CINORA-Inset, location claims from the nodes are distributed among a subset of cells to detect any replication. These cells are generated from a nonnull intersecting subset algorithm. The inherent property of this algorithm is for any two subsets C_i and C_j of total $1 \le i, j \le N$ cells, and $C_i \cap C_j \ne \emptyset$. Thus, during the authentication phase at least one cell receives conflicting location claims if adversary has ever attempted to replicate a legitimate node. In CINORA-Hybrid a base station-based two-phase authentication scheme is used in which a sensor node has a valid residence entry permit for a cell. If permitted and nodes current residing cell is different or two (or more) similar permits are detected with different location claims, then that identity node is removed from the network.

2.2.14. A Note-Based Randomized and Distributed Protocol for Detecting Node Replication Attacks in Wireless Sensor Networks. This technique falls into the category of witness node-based techniques. Meng et al. [53] have proposed a note-based randomized and distributed protocol called NRDP, for detecting node replication attacks, which introduces no significant overhead on the resource-constrained sensors. This protocol does not need the geographic locations of nodes as well. Three types of nodes are assumed in the network, namely, a *claimer node, a reporter node*, and *a witness node*. A node which broadcasts a claim message is a *claimer node*. Neighbor node which forwards a claim message is a *reporter node*. And the destination node of a claim message is a *witness node*. This protocol works in two phases: neighbor discovery period and replication detection period. In the beginning of NRDP, it is a neighbor discovery period in which each node in the network broadcasts a message within its one-hop neighbors. After neighbor discovery period, each node in the network gets a neighbor list. The replication detection period starts when the neighbor discovery period ends. Replication

detect period consist of two steps. The first step is called request-note step and the second step is called send claim step. In *request-note step*, node α randomly chooses a node γ from its neighbor list as its reporter node, and then sends a request-note message to the reporter node. Upon receiving α request-note message, node γ replies with a signature note message which contains a note. The parameter time is fresh time of the note. Nodes in the network use it to identify the validity of a note received in different iterations. Note is an evidence to prove that the reporter node of a claimer node is existing and valid. In the *send-claim step*, every node generates a claim message, which includes a signed subneighbor list and a note got from the corresponding reporter node. The parameter list in the claim message is an ID list, which consists of q α's neighbor node IDs. And the reporter node γ must be in the list. Each node α then broadcasts the claim message in one-hop neighbors. When the reporter node receives corresponding claim message, it first verifies the signature and the time fresh of the note contained in the claim message. Further, the reporter node verifies that the list in the claim message contains its ID. If all the verifications succeed, using a pseudorandom function, the reporter node calculates g witness nodes for the claimer node. This function takes in input, the ID of the claimer node, which is the first argument of the claim message, the current rand value, and the number g of witness nodes that has to be generated. The witness nodes of a certain node change in different iterations. A trusted entity broadcasts a seed *rand* to the network before each detection iteration starts. This prevents the adversary from anticipating the witness nodes in a given protocol iteration. The reporter node analyzes the claim message, then generates a forwarded claim message, and forwards the forwarded claim message to all the g witness nodes. The forwarded claim message just contains the subneighbor list signed by claimer node, without note. When a node receives a claim message, it first checks whether it is the corresponding reporter node. If it is the reporter node of the claimer node, it checks the signature, the fresh of the note, and the list in the claim message. If it is not the reporter node, with probability pc it does the checking jobs as the reporter node does. It is necessary for nonreporter node neighbors to do the checking jobs with probability pc. This can prevent a claimer node from specifying a nonexisting neighbor node as its reporter node. Each node in the network has to specify an actual neighbor node as its reporter node, or it will be detected as a replicated node by its neighbor nodes.

Each witness node that receives a forwarded claim message verifies the signature and time fresh firstly. Then, it compares the claim to each previously stored claim. If it is the first time received claim contains IDα, then it simply stores the claim. If a claim from IDα has been received, the witness checks whether the claimed neighbor list is the same as the stored claim. If a conflict is found, the witness detects a node replication attack. Then, the witness triggers a revocation procedure for IDα. Actually, because there is always only one reporter node for a claimer node, if the claimer node is a valid node, its corresponding witness nodes would never receive more than one forwarded claim message from the claimer node. Therefore, once a witness node receives two claims containing the same ID in one detection iteration, it detects a replication attack. The two signature claims become evidence to trigger the revocation of the replicated node. The witness node forwards both claims to the base station. The base station will broadcast a signature message within the network to revoke the replicated node.

2.2.15. Distributed Detection of Replication with Deployment Knowledge in Wireless Sensor Networks. This technique falls into the category of group-based techniques. Ho et al. [54] have proposed three group deployment knowledge-based schemes for the detection of node replication attack in wireless sensor networks. Their schemes are based on the assumption that nodes are deployed in groups. By taking advantage of group deployment knowledge, the proposed schemes perform replica detection in a distributed, efficient, and secure manner. The sensors can be preloaded with relevant knowledge about their own group's membership and all group locations. Then, the sensors in the same group should be deployed at the same time in the location given to that group. The three proposed schemes are basic, location claim, and multigroup approaches. The first scheme is the basic scheme in which each node only accepts the messages from the member's of their own group (trusted nodes) not from other groups (untrusted nodes). It stops intercommunication between groups. An advantage of this basic scheme is low communication and computational or memory overhead. But the problem that is even honest nodes suffer from communication due to the fact that the deployment points are far away from their group. The network becomes poorly connected and not suitable for high resilient applications. To solve this problem, second scheme is proposed which also forwards messages from untrusted nodes as long as they provide provable evidence that they are not replicas but based on only predetermined locations for replica detection. The second scheme achieves high replication detection capability with less communication, computational, and storage overheads as compared to the first scheme, but there is a risk of DoS by flooding fake claims.

The third scheme protects against this kind of aggressive adversary. Every sensor node sends its neighbor's location claims to multiple groups rather than a single group. This scheme has higher communication overhead. It can provide a trade-off between the overhead and resilience to attack. This scheme provides very strong resilience to node compromise, since attacker needs to compromise multiple groups of nodes to prevent replicas being undetected.

2.2.16. Distributed Detection of Node Replication Attack Resilient to Many Compromised Nodes in Wireless Sensor Networks. This technique falls into the category of group-based techniques. Sei and Honiden [55] have proposed a distributed protocol for the detection of node replication attack that is resilient to many compromised nodes. Their method does not need any reliable/trusted entities. To prevent an attacker from learning the location of a witness node of a compromised node, the protocol uses a one-time seed for each replicated node detection process; that is, each node has the role of

starting a detection process, and it is preloaded with the assigned turn number and seed for the turn.

When node has a turn starting detection process, it sends the seed and its ID with a signature. Other nodes verify the signature and execute the detection process if the verification succeeds. They divide nodes into groups to increase resiliency to fault nodes and compromised nodes. The role of the starting detection process is not assigned to each node but to each group. If at least one node of a group survives, the group can start the detection process during its turn. An attacker must compromise the first node of a group which has the next turn starting detection process if he wants to learn the location of the witness node in the next detection process.

2.2.17. A Resilient and Efficient Replication Attack Detection Scheme for Wireless Sensor Networks. This technique falls into the category witness node-based techniques. Kim et al. [56] have presented a distributed, deterministic approach to detect node replication attack. Their scheme works in three steps: initialization, witness node discovery phase, and node revocation phase. In initialization phase, before deployment, a base station (BS) associates a particular location coordinate (hereafter referred to as the verification point, vp) with each node id using geographic hash function F. A vp is the target location coordinate in the network where each sensor node will be verified, and it can be predetermined by a network operator to a certain extent with experience. In witness node discovery phase, the replicas with the same id but different deployment locations are detected through location claim message. In the last phase of node revocation base station BS floods the revocation node lists after checking out the revocation request message received from the witness nodes. Once a BS receives this revocation request message, it checks whether the revocation request message is correctly encrypted by witness node using a pair-wise key shared with witness node. If the key is correct, a BS floods a list of replica nodes including reporter node through the network. If the key fails, which means that an attacker sent the forged replica revocation message, the BS regards that reporter node has been compromised.

3. Comparison of Node Replica Detection Schemes for Static WSNs

In this paper, we have addressed an important attack on WSN referred to as node replication attack or clone node attack. So far, many techniques have been proposed to detect node replication attack in static WSNs which are broadly categorized into centralized and distributed techniques. We have compared all the techniques according to their year of publication, identifying their shortcomings.

3.1. Centralized Techniques. Centralized techniques are considered to be the first solutions for detecting replicated nodes which are simple but suffer from several common drawbacks. Some of the limitations of centralized techniques are found to be fairly serious like the base station which introduces a single point of failure, and any compromise of the base station will render the solution useless; also, even if there are no attacks the nodes surrounding the base station will suffer an undue communication burden which may shorten the lifetime of a network, and this approach also incurs an observable processing delay. Consequently, centralized detections have barely an advantage over distributed detections making a distributed solution a necessity. The asymptotic performance of centralized techniques (including their memory and communication cost) is shown in Table 1. Localized voting protocols are also considered as the first naïve solutions for the detection of clone nodes which are unable to deal with distributed node replication attacks, in which replicas are placed at least two hops away from each other. In order to detect replicas which are spreading anywhere in the network a fully distributed solution is needed that also incurs small memory and energy overhead.

In 2004, one of the first solutions for detecting replicated nodes was proposed by Dutertre et al., outlined in [57] which was based on a centralized base station for node replica detection. This scheme was the most straightforward one and a naive solution that provided a low defense against node replication attacks, suffering from several drawbacks as mentioned before.

In 2007, Brooks et al. [32] proposed a clone detection protocol which was based on random pairwise key predistribution schemes and used to tackle with detection of cloned cryptographic keys rather than clones sensor nodes. This solution seemed effective but only when the size of the keys predistributed to each node is small and more clones exist in the network, thus implying poor detection accuracy. Moreover, it is assumed in the protocol that the connections between all nodes are possibly equal, while practically in WSNs, any sensor node can only communicate with a limited number of neighbors within a finite wireless communication radius. Another drawback of this solution is that it has neglected to ensure that the participating clones report their keys honestly to the base station.

Choi et al. [23] proposed another centralized detection technique named SET in 2007 which was an attempt to reduce the detection overhead by computing set operations. But the message authentication codes used for additional security resulted in even higher detection cost in terms of computation and communication. Moreover, SET protocol is highly complex due to its complicated components, and unexpectedly an adversary can misuse the detection protocol to revoke honest nodes.

Another centralized approach was proposed in 2008 by Xing et al. [34] which used social fingerprint for the detection of clones, but it was purely based on fixed WSNs, and thus neither node addition nor disappearance can be handled. Furthermore, besides all the common limitations of centralized solutions, it cannot handle a sophisticated replica which can cleverly compute by itself a fingerprint consistent with its neighborhood in order to flee the detection at the sensor side. A more intelligent replica can dodge and avoid the detection at the base station simply by not communicating with the base station.

The most recent solution for the detection of node replication attack or clones is a centralized technique given

Table 1: Asymptotic performance of centralized schemes.

Type of scheme	Technique/scheme	Communication cost	Memory cost
Key usage based	Brooks et al. scheme [32]	$O(n \log n)$	—
Base station based	SET [23]	$O(n)$	$O(d)$
	CSI [38]	$O(n \log n)$	—
Neighborhood social signature based	Xing et al. scheme [34]	$C \cdot (1 + \text{ratio})$	$O(d) + \min(M, \omega \cdot \log_2 M)$
Cluster head-based techniques	Znaidi et al. scheme [36]	$O(t^2)$	$O(t)$

n: no. of nodes in the network, ω: the column weight in the superimposed s-disjunct code, C: message generated by sensor node, d: degree of neighboring nodes, M: the number of rows in the superimposed s-disjunct code, ratio = $\log_2 M / L_{\text{packet}} \times 100\%$, and L_{packet}: the bit-length of a regular message.

by Yu et al. [38] in 2012. They have used a novel concept of compressed sensing for the identification of clones in the sensor network. This technique has the lowest communication overhead, but it suffers from all the common drawbacks of centralized techniques as BS is responsible for the aggregation of the result (decision) about the identification of clones in the network.

Considering the limitations of centralized detection schemes, the researchers move to a distributed solution for detecting clones, and the first naïve solution that was proposed was called node-to-network broadcasting (N2NB). Although the scheme was simple it also suffered from high memory and communication cost for large sensor networks.

3.2. Distributed Techniques. We have investigated a dozen distributed detection protocols by asymptotically comparing their communication and memory costs, and they are shown in Table 2. As all the proposed solutions use different motivations and assumptions and thus have their respective strengths and weaknesses, we cannot make any general or definite remarks that which solution is the best one.

Distributed techniques for the detection of clone node attack are categorized into three main classes, namely, witness node-based, neighbor-based, and generation-based or group-based techniques. All the three categories have their own pros and cons. For neighbor-based technique [48], the neighboring nodes should be static and any addition or removal of nodes is not possible throughout the detection process because in doing so the detection process is affected severely. For the generation- or group-based techniques [40, 41, 54, 55] all the nodes are deployed in groups, and no new node can be added in a particular group. Also, nodes should have location or network information before node deployment. These techniques only prevent the node replication attack but are unable to detect the clone nodes.

The witness node-based techniques use a framework called claimer-reporter-witness framework in which a node referred to as claimer locally broadcasts it, location claim to its neighbors. Each neighbor serves as a reporter and employs a function to map the claimer ID, to a witness. The neighbor forwards the claim to the witness and if it receives two different location claims for the same nodded id then it means that the adversary has replicated a node. The adversary can also employ a function to know about the witness for the given claimer ID and may also locate and compromise the witness node before she inserts the replicas into the wireless sensor networks in order to evade the detection.

A relatively more mature distributed detection scheme was proposed in 2007 by Parno et al. [28] known as deterministic multicast (DM) which was the first to use a framework called claimer-reporter-witness framework. Although its design goal was to reduce communication cost, it was treated as an unfavorable protocol because of its several drawbacks. Firstly, it does not provide much security as an adversary only needs to compromise all the g witnesses for a given claimer id deploying as many replicas as she desires without activating an alarm. Secondly, it does not work for large g as both the network communication and the node storage are proportional to g, and with very small g, an adversary can produce unlimited replicas. Considering DM as unappealing due to its deterministic property, Parno et al. [28] have proposed and developed two more techniques as improvements of DM protocol, namely, randomized multicast (RM) and line selected multicast (LSM). The security was improved but at the price of increased communication/memory costs. In both of these protocols the problem lies in the selection of witness nodes (i.e., Probabilities) and also it is not always true that location claims of clone nodes are received to the same witness node. Moreover, both RM and LSM are unable to detect masked replication attack. To decrease the communication cost of RM protocol, LSM was developed as a less expensive version of RM, but it suffers from uneven distribution of witnesses nodes. As majority of witness nodes are selected from the center of the network, thus the energy of these nodes is depleted soon, and also they become the point of interest for the adversary.

Zhu et al. [44, 45] proposed two techniques called single deterministic cell (SDC) and parallel multiple probabilistic cells (P-MPC) in 2007 as the variations of DM. Practically, both of these techniques depend upon the careful selection of a cell size (s) because if the cell size is too large, they incur high communication cost like N2NB, and if s is too small, it will be very easy for an adversary to trounce them by compromising all nodes in the g deterministic tiny cells. An important problem with SDC is that in order to reduce the broadcast overhead, it requires to execute the flooding only when the first copy of a node location claim arrives at the cell, and the following copies are ignored. In doing this, the node in the cell that first receives the location claim is unable to distinguish between claims of original node and replica node.

Another attempt to detect clones was made by Conti et al. [42, 43] in 2007 who have proposed a randomized, efficient, and distributed protocol named RED by combining

Table 2: Asymptotic performance of distributed schemes.

Type of scheme	Technique/scheme	Communication cost	Memory cost
Node-to-network broadcasting	N2NB [28]	$O(n^2)$	$O(1)$
Witness node	DM [28]	$O(g \log \sqrt{n}/d)$	$O(g)$
	RM [28]	$O(n^2)$	$O(\sqrt{n})$
	LSM [28]	$O(n\sqrt{n})$	$O(\sqrt{n})$
	RED [42, 43]	$O(g \cdot p \cdot dn\sqrt{n})$	$O(g \cdot p \cdot d)$
	SDC [44, 45]	$O(r \cdot \sqrt{n}) + O(s)$	$O(\omega)$
	P-MPC [44, 45]	$O(r \cdot \sqrt{n}) + O(s)$	$O(\omega)$
	B-MEM [3]	$O(k \cdot n \cdot \sqrt{n})$	$O(tk + t'k\sqrt{n})$
	BC-MEM [3]	—	$O(tk + t'k\sqrt{n'})$
	C-MEM [3]	—	$O(t + t'\sqrt{n})$
	CC-MEM [3]	—	$O(t + t'\sqrt{n'})$
	Melchor et al. [51]	$O(\sqrt{n})$	$O(d)$
	RDE [52]	$O(d \cdot n \cdot \sqrt{n})$	$O(d)$
	RAWL [4]	$O(\sqrt{n} \log n)$	$O(\sqrt{n} \log n)$
	TRAWL [4]	$O(\sqrt{n} \log n)$	$O(1)^2$
	Kim et al. [56]	$O(\sqrt{n})$	$O(\sqrt{n})$
Generation or group based	Bekara and Laurent-Maknavicius [40, 41]	$O(\sqrt{n})$	$O(1)$
	Basic scheme [54]	$O(m)$	$O(m)$
	Location claim base scheme [54]	$O(m + d)$	$O(d + 2m)$
	Multigroup base scheme [54]	$3 * O(m + d)$	$O(d + 2 * m\,(1 + D_{\max}))$
	Sei and Honiden [55]	$O(r)$	$O(r \cdot \sqrt{n})$
—	Ho [49]	$O(n\sqrt{n})$	$O(n)$
Neighborhood based	NBDS [48]	$O(r \cdot \sqrt{n})$	$O(r)$

n: no. of nodes in the network, d: degree of neighboring nodes, g: no. of witness nodes, r: communication radius, s: the number of sensors in a cell, p: probability that neighboring node will forward the location claim, ω: the column weight in the superimposed s-disjunct code, and ξ: distinct IDs from set of nodes as monitor.

the benefits of both DM and RM. This protocol is considered to be the most promising detection protocol which has solved the crowded center problem as the selection of witness nodes is random and fully distributed. Also, RED [4] is such an "area oblivious" protocol that associates sensor nodes with almost even responsibility, and the selection of witness nodes is pseudorandom which leads to a uniform witness distribution. Besides these advantages, the only drawback of RED is the deterministic selection of witness nodes and that the infrastructure for distributing RED's random seed may not always be available. RED is also unable to detect masked replication attack.

Bekara et al. [40, 41] in 2007 proposed a solution for preventing WSN from node replication attack which exploits the fact that excluding new nodes from joining the network can prevent replication attacks. The main drawback of this scheme is that the sensor nodes are bound to their groups and geographic locations.

In 2009, Zhang et al. [3] have proposed four memory efficient multicast protocols for the detection of replicated nodes, namely, Bloom filter MEM, Bloom filters and cell forwarding MEM, cross forwarding MEM, and last is cross forwarding and cell forwarding MEM. B-MEM is an extension of LSM, but it incurs additional memory consumption per node, and it may also lower the detection rate of LSM due to false verifications (false positives of Bloom filters). BC-MEM requires highly accurate localization due to its cell division and anchor node selection which may not be affordable for current generation of WSNs. Also, an adversary can elude BC-MEM by compromising certain deterministic anchor nodes. In case of both C-MEM and CC-MEM, cross forwarding achieves high detection probability for convex deployment field (particularly for rectangle-shaped deployment field), but for other irregular topologies considered by LSM (like thin cross and large H), these two schemes may work poorly by dropping the detection rate significantly.

A simplified version of N2NB was proposed by Zhang et al. [3] in 2009 known as randomly directed exploration (RDE). Its network communication overhead is reduced, but storage cost remains the same with N2NB. The detection rate is also decreased and may not be very significant even for a convex deployment field concluding that RDE appears to be feasible only for an ideal network model.

Another work in this area is done by Zeng et al. [4] in 2010 who have proposed two detection protocols, namely, RAndom WaLk (RAWL) and Table-assisted RAndom WaLk (TRAWL) for the detection of node replication attack. Both of these protocols are an extension of LSM and thus suffer from the same drawbacks. Although they have much higher detection probability than LSM, both RAWL and TRAWL require more than twice the communication overhead of LSM.

For an inclusive survey, we have also analyzed some other distributed techniques which are neither very popular nor have promising results in detecting node replication attack. These techniques include Ho et al. [54] proposed in 2009, Kim et al. [56] proposed in 2009, and Meng et al. [53] proposed in 2010.

4. Detection Techniques for Mobile WSNs

Mobility has become an important area of research for WSN community. In mobile WSNs, mobility plays a key role in the execution of the application as the introduction of mobile entities can resolve some problems and offer many advantages over the static WSNs. The node replica detection techniques developed for static WSNs, do not work when the nodes are expected to move as in mobile WSNs, and thus they have turned out to be ineffective for mobile WSNs. As a result some techniques (still not mature enough) have also been developed for mobile WSNs to detect the replica or clone nodes. These techniques are classified into two main classes as centralized and distributed and are described below.

4.1. Centralized Techniques

4.1.1. Fast Detection of Replica Node Attack in Mobile Sensor

Networks Using Sequential Analysis. Ho et al. [58, 59] have proposed a mobile replica detection scheme based on the sequential probability ratio test (SPRT) [50]. Their protocol is based on the fact that an uncompromised mobile node should never move at speeds in excess of the system-configured maximum speed. As a result, an uncompromised (original) mobile sensor node measured speed will appear to be at most the system-configured maximum speed as long as speed measurement system with low error rate is employed. On the other hand, replica nodes will appear to move much faster than original nodes, and thus their measured speeds will likely be over the system-configured maximum speed because they need to be at two (or more) different places at once. Accordingly, if it is observed that a mobile node measured speed is over the system-configured maximum speed, it is then highly likely that at least two nodes with the same identity are present in the network. By leveraging this intuition, the SPRT is performed on every mobile node using a null hypothesis that the mobile node has not been replicated and an alternate hypothesis that it has been replicated. In using the SPRT, the occurrence of a speed that either lessens or exceeds the system-configured maximum speed will lead to acceptance of the null and alternate hypotheses, respectively. Once the alternate hypothesis is accepted, the replica nodes will be revoked from the network.

4.1.2. A New Protocol for the Detection of Node Replication

Attacks in Mobile Wireless Sensor Networks. Deng and Xiong [60] have proposed a new protocol to detect the replicas in mobile WSNs. They have used the idea of polynomial-based pair-wise key pre-distribution and Bloom Filters which insure that the replicas can never lie about their real identifiers and collect the number of pair-wise keys established by each sensor node. Replicas are detected by looking at whether the number of pair-wise keys established by them exceeds the threshold. The protocol works in three steps, node initialization, pair-wise establishment, and detection. In node initialization, before nodes are deployed, the key server randomly generates a bivariate symmetric polynomial over a finite field. After deployment between nodes, pairwise keys are established. Each node periodically constructs a report, which includes its ID and counting Bloom filter (or compressed counting Bloom filter), and sends it to the base station. At base station, counting bloom filters collect the number of pairwise keys established by each node. Nodes whose number of pair-wise keys exceeds the threshold value are considered to be the clones.

4.2. Distributed Techniques

4.2.1. Mobile Sensor Networks Resilient against Node

Replication Attacks. Chia et al. [61] proposed a novel protocol, called extremely efficient detection (XED), against node replication attack in mobile sensor networks. The idea behind XED is motivated from the observation that for the networks without replicas, if a sensor node s_i meets the other sensor node s_j at earlier time and s_i sends a random number r to s_j at that time, then when s_i and s_j meet again, s_i can ascertain whether this is the node s_j met before by requesting the random number r. Based on this observation, a "remember and challenge strategy" is proposed. Once two sensor nodes, s_i and s_j, are within the communication ranges of each other, they first, respectively, generate random numbers $rs_i \rightarrow s_j$ and $rs_j \rightarrow s_i$ of b bits, and then they exchange their generated random numbers. They also use a table to record the node ID, the generated random number, and the received random number in their respective memory. In case the pair of two nodes met before, the above procedure is also performed such that the random number stored in the memory is replaced by the newly received random number. Consider the example shown in Figure 4, in which the sensor node s_i meets another sensor node s_j. If s_i never meets s_j before, they exchange random numbers. Otherwise, the sensor node s_i requests the sensor node s_j for the random number $rs_i \rightarrow s_j$ exchanged at easier time. For the sensor node s_i, if the sensor node s_j cannot replies or reply a number which does not match the number in s_i memory, s_i announces the detection of a replica. When the replicas meet the genuine nodes, the replicas can always pretend that they meet for the first time. However, if the genuine nodes have a record showing that they ever met at earlier time, the replicas are also detected.

4.2.2. Efficient and Distributed Detection of Node Replication

Attacks in Mobile Sensor Networks. Chia et al. [62] proposed an efficient and distributed detection (EDD) scheme and its variant, storage-efficient EDD (SEDD) scheme to detect the node replication attack. The idea behind EDD and SEDD is motivated from the following observations. For a network without replicas, the number of times, $\mu 1$, in which the node u encounters a specific node v, should be limited in a given time

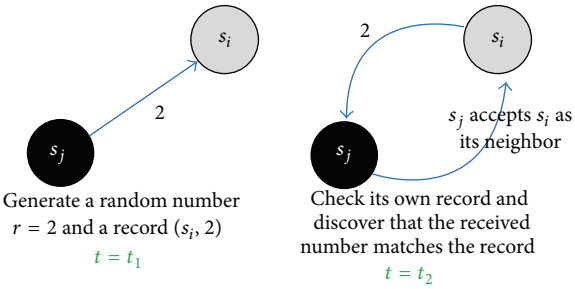


FIGURE 4: The operations between two genuine nodes in XED at time t_1 and t_2 (gray and black nodes are genuine) [61].

interval of length T with high probability. For a network with two replicas v, the number of times, $\mu 2$, in which u encounters the replicas with the same ID v, should be larger than a threshold within the time interval of length T. According to these observations, if each node can discriminate between these two cases, each node has the ability to identify the replicas. The EDD scheme is composed of two steps: offline step and online step. The offline step is performed by the network planner before the sensor deployment. The goal is to calculate the parameters, including the length T of the time interval and the threshold ψ used for discrimination between the genuine nodes and the replicas. On the other hand, the online step will be performed by each node per move. Each node checks whether the encountered nodes are replicas by comparing ψ with the number of encounters at the end of a time interval. It can be observed from EDD that each node should maintain a list L, leading to $O(n)$ storage overhead. A storage-efficient EDD (SEDD) scheme is proposed based on the tradeoff between storage overhead and time interval length. The basic idea behind SEDD is that instead of monitoring all nodes, each node only monitors a subset of nodes, called monitor set, in a specific time interval. When the cardinality of the monitor set is selected as ξ, the simplest way for each node to select the nodes to be monitored at the beginning of a time interval is to randomly pick ξ distinct IDs from $\{1, \ldots, n\}$. Since the storage overhead is equal to the number of nodes being monitored, the storage overhead is reduced to the cardinality of the monitor set, $O(\xi)$, in the SEDD scheme.

4.2.3. Patrol Detection for Replica Attacks on Wireless Sensor Networks.

Wang and Shi [63] have employed mobile nodes as patrollers to detect replicas distributed in different zones in a network, in which a basic patrol detection protocol and two detection algorithms for stationary and mobile modes are presented. The detection of replicas in stationary sensors is based on the assumptions that if two or more sensors in different locations have the same ID, then all the nodes with the ID will be regarded as compromised nodes or its replicas. Also, for mobile sensors (patroller), if a mobile node moves with a speed higher than the denoted maximum speed, it will be regarded as a replica attack. In the replica detection of static sensor nodes, when a mobile patrol node moves to a new zone, it first discovers its location and then broadcasts its patrol claim. Each node will be patrolled by at least two mobile nodes. After receiving the location messages, the stationary node takes the mobile nodes who patrolled him as the anchor nodes and will send the patrol node its location claim. After collecting the answer message, patrol node will check the location of node, and if the distance is larger than the signal range, it ignores the wrong message. Otherwise, it will check the ID of the answer message by using the security assumption "A legitimate ID only has one location." Then, it saves the answer from the original node (benign node) in a whitelist, saves the replica node ID in a blacklist, and revokes the replica ID by refusing to distribute secret material and broadcasting its two answer messages to other mobiles nodes. Then, patrol node will move to another location to send his patrol claim in another interval. After a round, it collects all the saved information of the white- and blacklists to the user when collecting the sensing data. If the replicas are deployed in a zone where a patrol node collects their answer message in a patrol interval, then the patroller can revoke them immediately after he receives the second answer and the distance between the two locations exceeds. Else if the replicas' answers are collected by different patrol nodes, then they will be found by the base station or by exchange messages of patrollers after a round. If the adversary compromises and replicates the patrol node, firstly, an original mobile patroller will wait for the answer message after he reaches a new position and sends his claim in time T, so there is a static period interval after the patrol broadcasts his claim. Accordingly, if the patroller node moves and changes its position in time $(T, T + \text{interval})$, then it is highly likely that at least two nodes with the same identity are present in the network. Further, the mobile patroller should never move faster than the system-configured maximum speed $V_{\max}$.

4.2.4. Single-Hop Detection of Node Clone Attacks in Mobile Wireless Sensor Networks.

Lou et al. [64] have proposed a node clone attack detection protocol, namely, the single hop detection (SHP) for mobile wireless sensor networks. The SHD protocol exploits the fact that at any time, a physical node (or equivalently, its node ID and private key) cannot appear at different neighborhood community; otherwise, there must be replicas in the network. The neighborhood community of a node is characterized by its one-hop neighbor node list, which is readily available in a typical WSN since sensor nodes need to know their neighbors in order to communicate with each other. The SHD protocol consists of two phases, the fingerprint claim and the fingerprint verification phases. In the fingerprint claim phase each node is required to sign its neighbor node list. The signed neighbor node list is a fingerprint of its current neighborhood community, hereafter referred to as fingerprint claim. The fingerprint claim is broadcasted in one-hop neighborhood. Upon reception of a fingerprint claim from a neighboring claim node, the receiver node will decide whether to become a witness node of the claim node. When it decides to become a witness node, the node will then verify the fingerprint claim and finally store the fingerprint claims of the witnessed nodes locally if the claim passed the verification process. In the fingerprint

verification phase, when two nodes meet with each other, they exchange their witnessed node lists, and this can be done by piggybacking the witnessed node list in the two nodes and then checking for a possible fingerprint claim conflict with received claims. In a fingerprint claim conflict, there are two fingerprint claims with the same ID and private key claiming two different neighborhood communities, which implies two detected replicas.

4.2.5. Detecting Node Replication Attacks in Mobile Sensor

Networks: Theory and Approaches. Zhu et al. [65] have proposed two replica detection algorithms for mobile sensor networks. First algorithm is a token-based authentication scheme proposed for the detection of replication attack in which the replicas do not cooperate (nonconspiring case). For the case in which the replicas cooperate by communicating with each other in an efficient manner, a detection method is proposed which is based on statistics and the random encounters between physical nodes. In the first algorithm, the base station periodically broadcasts to the entire sensing region a timestamp protected by a broadcast authentication protocol. The broadcast announces the beginning of a detection round. Upon hearing the timestamp, a genuine mobile node randomly selects a secret seed $s_i \in \{0,1\}^l$, where l is a common security parameter, and empties its local storage of the previously received tokens. The detection consists of a token exchange phase and a mutual authentication phase When a mobile node first meets with another mobile node in the detection round, they will exchange a token with each other and will record the tokens in their memories. When these mobile nodes meet again in the same detection round, each will ask the other for the previously exchanged token. Upon receiving the correct reply, each believes that the other is authenticated. Otherwise, in case of replica, when genuine node asks a replica node (to whom it met before) about the token they have exchanged in their first meeting, the replica node will reply in no or with a wrong token which will mark him as replica.

The second algorithm is a statistics-based detection scheme for detecting replicas that cooperate with each others. This idea is partially inspired by [66] whose detection principle is that if a node is not "seen again" by others, it is likely that the node has been captured. Similarly, herein, the principle is that if in a certain detection round a node is "seen again" too many times by others, it is likely that the node is a replica. Every genuine node contains a step counter "T" and also its "acquaintance list" consisting of n Boolean variables. Each time a mobile node meets another mobile node, it increases the counter T by 1. If this is its first meeting with any mobile node, it treats it as an acquaintance and sets the corresponding bit in the list to 1. Once the acquaintance list contains all 1's, the statistics stops. In the nondetection stage each node reports its numbers of meetings with others when dropping by the base station. The base station is employed for centralized analysis. Finally, the node with more encounters is detected as replica, and base station finally broadcasts the entire network for replicated IDs.

4.2.6. Emergent Properties: Detection of the Node Capture

Attack in Mobile Wireless Sensor Networks. Conti et al. [66] have proposed two algorithms for the detection of node capture attack in mobile wireless sensor networks. Their first algorithm is simple distributed detection (SDD) in which the attack is detected using only information local to the nodes. The second algorithm is called cooperative distributed detection (CDD) which exploits node collaboration to improve the detection performance. Both of the proposed algorithms are based on the simple observations that if node a will not remeet node b within a certain period of time, then it is possible that node b has been captured. Hence, node a can autonomously know the probability that a "not yet remet" node has been actually captured by the adversary. The SDD follows the above simple observation that each node a is given the task of tracking a specific set T_a of other nodes. For each node $b \in T_a$ that a gets into the communication range of, a set the corresponding meeting time to the value of its internal clock and start the corresponding timeout, that will expire after λ seconds. If the time-out expires (i.e., a and b did not remeet), the network is flooded with an alarm triggered by node a to revoke node b. In CDD, network mobility and node cooperation are leveraged to improve node capture detection. When two nodes a and b exchange information about the nodes (if any) that are tracked by both a and b, that is, the nodes in T_aT_b, the node exchanges information only when cooperating nodes are in the same communication radius. This shared information is further used for node capture.

4.2.7. Mobility-Assisted Detection of the Replication in Mobile

Wireless Sensor Networks. Deng et al. [67] have proposed two schemes for the detection of node replication attack in mobile wireless sensor networks. The first is called unary time location storage and exchange (UTLSE), and, second is called multitime location storage and diffusion (MTLSD). In both protocols, after receiving the time-location claims, witnesses carry these claims around the network instead of transmitting them. That means that data are forwarded only when appropriate witnesses encounter each other. Only if two nodes encounter each other, they exchange their time-location claims, that is, if a tracer receives a time location claim from its tracked neighbor node, it does not immediately transmit this time-location claim to the witness if the witness is not currently within its communication range but stores that location claim until encountering the witness. UTLSE detects the replicas by each of the two encountered witnesses which stores only one time-location claim. On the other hand, MTLSD stores more time-location claims for each tracked node and introduces time-location claims diffusion among witnesses. The detection probability of the MTLSD protocol is greater than the probability of protocol UTLSE.

5. Comparison of Node Replica Detection Schemes for Mobile WSNs

Mobile wireless sensor networks (MWSNs) are still in their infancy, and there are many challenges in MWSNs that are

still needed to be resolved. These challenges include deployment, localization, self-organization, navigation and control, coverage, energy, maintenance, and data process [26]. In case of localization, node position can be determined once during initialization when sensor nodes are deployed statically [68]. However, when sensor nodes are mobile, they must continuously obtain their positions as they navigate through the whole sensing region. As a result, in mobile WSNs, localization requires additional time and energy and also the availability of a rapid localization service. Due to the dynamic network topology of mobile WSNs, they cannot rely on routing tables or recent route histories as static WSNs do for passing messages through the network because table data become outdated quickly; thus, route discovery data must repeatedly be performed extensively in terms of power, time, and bandwidth.

Ho et al. [58, 59] have proposed a centralized detection scheme for mobile WSNs in which accurate measurement is a prerequisite for acceptable false-negative and -positive rates. In result, it requires dynamic and precise localization system and a tight time synchronization which are both nontrivial tasks. Also, better and accurate sampling entails even much more expensive equipment (GPS) and thus may not be affordable for the current generation of WSNs. Another centralized detection technique is proposed by Deng and Xiong [60] in which there is no way to ensure, the participating clone node will report their keys honestly to the base station. It is possible that an original node number of pairwise keys exceed the threshold value due to its communication. Also as the effectiveness of both the above centralized detection techniques relies on the involvement of the base station, this easily incurs the problems of single-point failure and fast energy depletion of the sensor nodes around the base station.

Yu et al. [61] have proposed distributed detection technique called extremely efficient detection technique (XED) in which the authors have assumed that the replicas cannot communicate and collaborate (or cooperate) with each other which is the weakness of this scheme because in case when the replicas cooperate with each other, they can establish secret channels among each other, and then they can easily deceive the detection technique. Efficient and distributed detection (EDD) is another distributed detection technique for mobile WSNs proposed by Yu et al. [62] which is inapplicable due to high storage overhead for large-scale WSNs.

Zhu et al. [65] have proposed a token-based detection technique which fails when a smart attacker establishes secret channels among replicas as by doing this, replicas can share the tokens and make the protocol exist in name only.

Conti et al. [66] have proposed two solutions, namely, SDD and CDD for the detection of node capture. Their approach is based on a simple observation which completely assumes that there is no membership change in the network; for example, at least no nodes die out (meaning run out of power) which is not the case in reality. Also, it is assumed implicitly that any senor node is able to flood the entire mobile WSN with a broadcast message which is also not possible in reality.

An asymptotic comparison of all the detection schemes for mobile WSNs is shown in Table 3 where their communication and memory costs are compared. As all the proposed solutions use different motivations and assumptions and thus have their respective strengths and weaknesses, we cannot make any general or definite remarks that which solution is the best one.

6. Discussion

Node replication attack or clone attack is one of the most harmful and dangerous threat to an unattended wireless sensor network because in this attack an adversary not only compromises the sensor nodes but can also carry out a large class of internal attacks for instance DoS attack, Sybil attack, and Black hole, and wormhole attack, by surreptitiously inserting arbitrary number of replicas at strategic positions of the network. Furthermore this is more niggling and troublesome because these replicated nodes, under the control of an adversary, having all the keying materials, pretend as authorized users in the network and thus deceiving the network into accepting them as legitimate nodes. It is difficult to identify replicas because of two major reasons. First, since a clone or replica is considered to be completely honest by its neighbors, the legitimate nodes cannot be aware of the fact that they have a clone among them. Voting mechanisms [33, 69] remain unsuccessful to detect clone nodes that are not within the same neighborhood as a voting mechanism is used to detect misbehaving nodes and clones within the neighborhood to agree on the legitimacy of a given node. Thus, there is a need for global countermeasure that can detect clones on the global level. Second, the general purpose security protocols for secure sensor network communication would allow replica nodes to create pair-wise shared keys with other nodes and the base station, and thus in doing so, the replica nodes are able to encrypt, decrypt, and authenticate all of their communications as if they were the original captured nodes.

The process or stages of node replication attack can be described in the form of a flow chart as shown in Figure 5. The flow chart concisely describes the instigation of node replication attack and its detection, from physical node capture, extraction of secret credentials, cloning and redeployment and finally the detection and prevention of node replication attack. At Stage 1, an adversary physically captures a sensor node. After physical capture the sensor node remains absent from the network for a specific period of time. If this absence of a sensor node is detected or a tamper-proof hardware is used, the attack will be prevented. Otherwise, an attacker or an adversary starts extracting all the secret materials of the captured node at Stage 2. At Stage 3, an adversary reprograms the captured node. If an adversary is unable to use a new hardware, it can compromise the node and then exploits the compromised node to disrupt the network operations by its misbehaving activities. At Stage 4, an adversary makes clones or replicas of the captured nodes by using new hardware, and these replicas have the same ID and all other keying materials as that of the captured node. After making clones or replicas, an adversary redeploys them

Table 3: Asymptotic performance of schemes against clone node attack in mobile sensor networks.

Nature of scheme	Type of scheme	Technique/scheme	Communication cost	Memory cost
Centralized	Node speed based	Ho et al. scheme [58, 59]	$O(n\sqrt{n})$	$O(n)$
	Key usage based	Deng and Xiong scheme [60]	$O(n \log n)$	—
Distributed	Information exchange based	XED [61]	$O(1)$	$O(4 \cdot d \cdot E[X])$
	Node meeting based	EDD [62]	$O(1)$	$O(n)$
		SEDD [62]	$O(n)$	$O(\xi)$
	Mobility assisted based	Wang and Shi scheme with Base Station [63]	$O(n)$	—
		Wang and Shi scheme with out Base Station [63]	$O(n * \sqrt{k})$	—
		UTLSE [67]	$O(n)$	$O(\sqrt{n})$
		MTLSD [67]	$O(n)$	$O(\sqrt{n})$

n: no. of nodes in the network, ξ: distinct IDs from set of nodes as monitor, d: degree of neighboring nodes, and k: total number of zones.

at strategic positions of the network for further insider attacks at Stage 5. Finally these replicas or clones can be detected by using various detection schemes.

Since clone nodes carry all the cryptographic and keying materials, all the traditional authentication and intrusion detection techniques are ineffective to discover and detect these clones or replicas in the network. Keeping this in mind many techniques have been proposed for the detection of node replication attack and recall that these are broadly categorized into centralized and distributed techniques. Some fairly serious limitations of centralized technique like the base station introduces a single point of failure, and any compromise of the base station will make the solution useless thus making distributed solutions a necessity. One important class of distributed techniques is witness node-based techniques which are considered to the most favorable techniques yet for detecting clone nodes. But according to Zeng et al. [4], replica detection protocols must be non-deterministic and fully distributed in order to circumvent the existing drawbacks of witness-based strategies. The witness node-based strategies ought to fulfill three requirements to have a high probability of detecting clones or replicas. Firstly, the selection of witness-nodes should be nondeterministic as it is more difficult for an adversary to launch clone attacks in nondeterministic protocols successfully because the witnesses of node are not known and are different in each execution of the protocol. Secondly, for any given node, all the nodes should have an equal probability to be the witnesses of that node during the lifetime of the network. Thirdly, the witness-nodes should be selected from all over the network randomly and not from particular area of the network every time meaning that the witness distribution should be uniform throughout the entire network.

There are two types of attacks which are the variations of node replication attack and can be launched by an adversary against witness node-based schemes. These are named as smart attack and masked replication attack. Smart attack is a special witness compromising attack, and in this attack an adversary avariciously chooses which sensor to corrupt in order to maximize its chance for its replicas to go undetected. The adversary finds out the witness nodes which are used to detect replicas and only compromises these witness nodes to avoid detection. The witness node-based techniques use a framework called claimer-reporter-witness framework in which a node referred to as claimer, locally broadcasts it location claim to its neighbors. Each neighbor serves as a reporter and employs a function to map the claimer ID to a witness. The neighbor forwards the claim to the witness and if it receives two different location claims for the same node ID then it means that the adversary has replicated a node. The adversary can also employ a function to know about the witness for the given claimer ID, and may also locate and compromise the witness node before she inserts the replicas into the wireless sensor network in order to evade the detection. In masked replication attack, the adversary may turn to compromise all the neighbors of a replica so as to prevent a location claim from propagating to any witness thus eliminating the reporters at all. This attack makes it possible for such a replica, whose neighbors have all been compromised, to lie about its physical position. So far, all the witness node-based techniques have assumed a static WSN, and are seemed to be the most promising schemes till yet to detect replicas or clones in static WSN, but alas these witness node-based schemes and location-based replication detection schemes are unable to detect and counter these types of replication attacks.

Nowadays, mobility has become an important area of research for WSN community. In mobile WSNs, mobility plays a key role in the execution of the application [68] as the integration of mobility in WSN can improve the coverage and utility of the sensor network deployment and enables more versatile sensing applications as well. However, besides that the introduction of mobile entities (which freely roam in the network and are autonomous as being able to reposition and organize themselves in the network) can resolve some problems by offering many advantages over the static WSNs the unique properties of mobile WSNs and the dynamic mobile network topology pose many new challenges in the security of mobile WSNs. The idea of detecting clone nodes in static WSNs is extensively based on the elitism of the node location meaning that a sensor node should be allied to a unique deployment position, and if one logical node id is found to be associated with two or more physical locations, the node replication is detected. But noticeably this is not

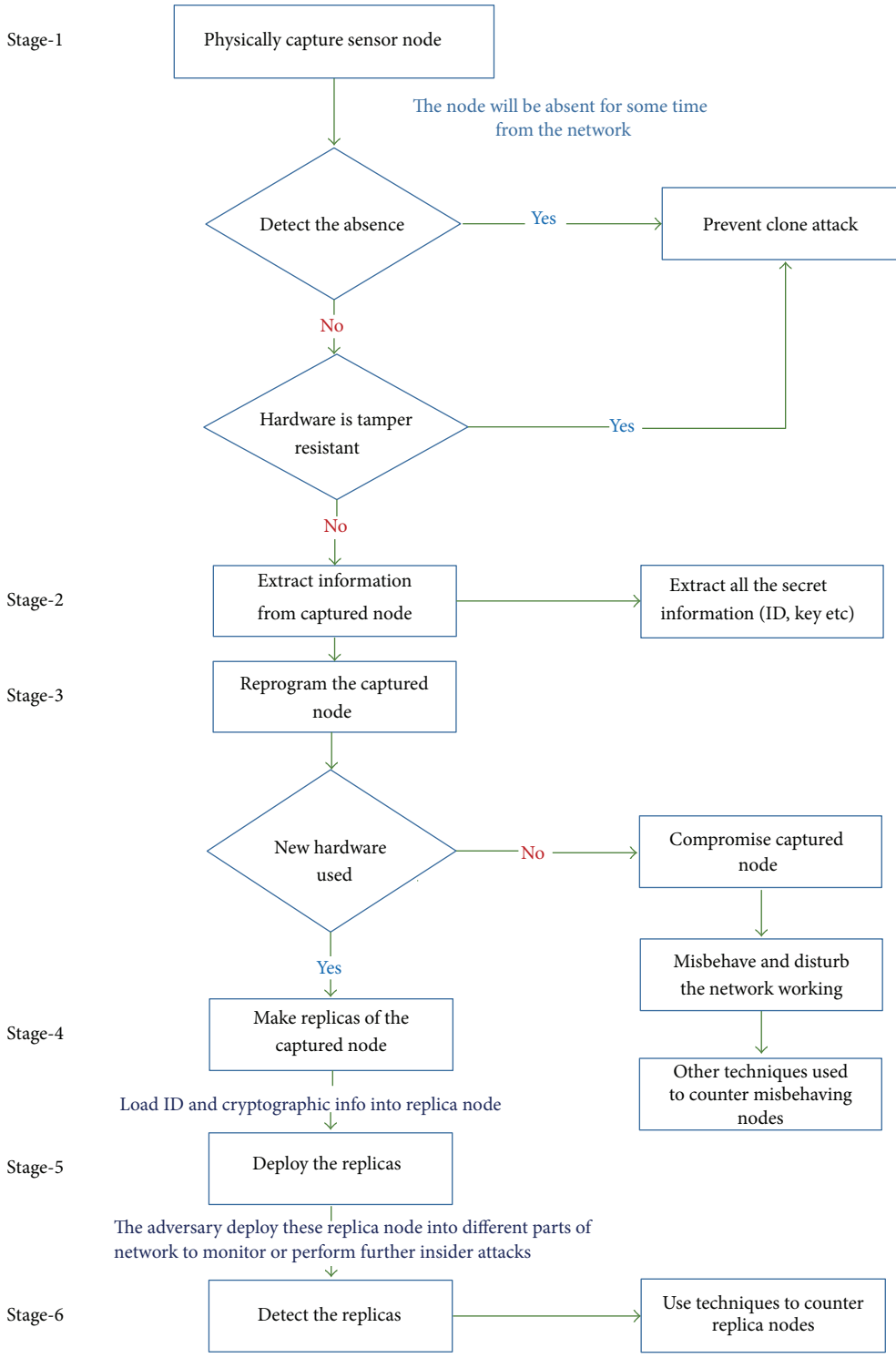


Figure 5: Stages of node replication attack in wireless sensor networks.

applicable to the emerging mobile WSNs where the sensor nodes are moving freely all the time in the network. Thus, a little work (which includes significantly different scenarios and techniques) has been done so far to deal with replicas or clones in mobile WSNs.

In mobile WSNs, the adversary is also mobile. In the literature, the assumed scenario of mobile WSN is that the sensors are unable to transmit sensed data at their will because the sink is not always present. Thus, the data accumulated in their memories become targets of many adversaries. In

[30], a mobile adversary model is proposed in which mobile adversary visits and travels around the network trying to compromise a subset of sensors within the time interval when sinks are not present in the network. The time taken by a mobile adversary to compromise a set of sensors is much shorter than the time between two successive data collections of a sink. Thus, it is much difficult to snatch mobile compromised nodes as well as mobile clones.

Another challenge arises in mobile WSNs when a mobile adversary adopts a more sophisticated strategy named "group mobility strategy." In this stratagem, the replicas form a physically close group which always moves together, but only a representative of them communicates with the genuine nodes, whereas the rest of the replicas remain inactive as "silent learners" so that they can learn (from encounters with genuine nodes) about any received token or corresponding meeting instant. Once the replicas have met all the "n" genuine nodes (and thus acquired all the necessary knowledge to pass later authentications), they can scatter in the sensing region, and each behaves actively and independently, until the next detection round starts.

Also, when the mobile replicas communicate and collaborate with each other and share their keys or random numbers, they can make the detection technique fails to thwart them easily. Thus, mobile WSNs offer much more challenges in detecting mobile replicas, and it is highly needed to overcome these challenges by developing some new, different and more efficient detection techniques for detecting mobile replicas or clones.

7. Conclusion

This paper reviewed the state-of-the-art schemes for detection of node replication attack also called clone attack. The existing techniques are broadly categorized into two classes distributed and centralized. Both classes of schemes are proficient in detecting and preventing clone attacks, but both schemes also have some noteworthy drawbacks. However, to sum up, the current study highlights the fact that there are still a lot of challenges and issues in clone detection schemes that need to be resolved to become more applicable to real-life situations and also to become accepted by the resource constrained sensor node.

Acknowledgment

The authors wish to acknowledge the anonymous reviewers for their valuable comments for the improvement of this paper.

References

[1] T. Bonaci, P. Lee, L. Bushnell, and R. Poovendran, "Distributed clone detection in wireless sensor networks: an optimization approach," in *Proceedings of the 2nd IEEE International Workshop on Data Security and Privacy in Wireless Networks (WoWMoM '11)*, Lucca, Italy, June 2011.

[2] W. T. Zhu, J. Zhou, R. H. Deng, and F. Bao, "Detecting node replication attacks in wireless sensor networks: a survey," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 1022–1034, 2012.

[3] M. Zhang, V. Khanapure, S. Chen, and X. Xiao, "Memory efficient protocols for detecting node replication attacks in wireless sensor networks," in *Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP '09)*, pp. 284–293, Princeton, NJ, USA, October 2009.

[4] Y. Zeng, J. Cao, S. Zhang, S. Guo, and L. Xie, "Random walk based approach to detect clone attacks in wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 5, pp. 677–691, 2010.

[5] R. Di Pietro, L. V. Mancini, C. Soriente, A. Spognardi, and G. Tsudik, "Data security in unattended wireless sensor networks," *IEEE Transactions on Computers*, vol. 58, no. 11, pp. 1500–1511, 2009.

[6] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

[7] B. Parno, M. Luk, E. Gaustad, and A. Perrig, "Secure sensor network routing: a cleanslate approach," in *Proceedings of the ACM CoNEXT Conference (CoNEXT '06)*, December 2006.

[8] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," in *Proceedings of the IEEE INFOCOM*, 2004.

[9] L. Yu and J. Li, "Grouping based resilient statistical en-route filtering for sensor networks," in *Proceedings of the IEEE INFOCOM*, 2009.

[10] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 259–271, May 2004.

[11] H. Chan, A. Perrig, and D. Song, "Secure hierarchical in-network aggregation in sensor networks," in *Proceedings of the13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 278–287, November 2006.

[12] J. Deng, R. Han, and S. Mishra, "Security support for in network processing in wireless sensor networks," in *Proceedings of the ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN '03)*, pp. 83–93, 2003.

[13] B. Przydatek, D. Song, and A. Perrig, "SIA: secure information aggregation in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 255–265, November 2003.

[14] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: a secure hop-by-hop data aggregation protocol for sensor networks," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '06)*, pp. 356–367, May 2006.

[15] S. Capkun and J. P. Hubaux, "Secure positioning in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 221–232, 2006.

[16] S. Ganeriwal, S. Čapkun, C. C. Han, and M. B. Srivastava, "Secure time synchronization service for sensor networks," in *Proceedings of the ACM Workshop on Wireless Security (WiSe '05)*, pp. 97–106, September 2005.

[17] X. Hu, T. Park, and K. G. Shin, "Attack tolerant time synchronization in wireless sensor networks," in *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM '08)*, pp. 41–45, Phoenix, Ariz, USA, April 2008.

[18] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks,"

in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 91–98, April 2005.

[19] D. Liu, P. Ning, and W. Du, "Attack-resistant location estimation in sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 99–106, April 2005.

[20] H. Song, S. Zhu, and G. Cao, "Attack resilient time synchronization for wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 112–125, 2007.

[21] K. Sun, P. Ning, C. Wang, A. Liu, and Y. Zhou, "TinySeRSync: secure and resilient time synchronization in wireless sensor networks," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 264–277, 2006.

[22] C. Hartung, J. Balasalle, and R. Han, "Node compromise in sensor networks: the need for secure systems," Tech. Rep. CU-CS-988-04, Department of Computer Science, University of Colorado at Boulder, 2004.

[23] H. Choi, S. Zhu, and T. F. L. Porta, "SET: detecting node clones in sensor networks," in *Proceedings of the 3rd International Conference on Security and Privacy in Communication Networks (SecureComm '07)*, pp. 341–350, September 2007.

[24] S. Gautam Thakur, "CINORA: cell based identification of node replication attack in wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications Systems (ICCS '08)*, 2008.

[25] S. Hussain and M. S. Rahman, "Using received signal strength indicator to detect node replacement and replication attacks in wireless sensor networks," in *Data Mining, Intrusion Detection, Information Security and Assurance, and Data Networks Security 2009*, vol. 7344 of *Proceedings of SPIE*, April 2009.

[26] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[27] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *International Journal of Computer and Telecommunications Networking*, vol. 38, no. 4, pp. 393–422, 2002.

[28] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in *Proceedings of the IEEE Symposium on Security and Privacy (IEEE S and P '05)*, pp. 49–63, May 2005.

[29] A. Seshadri, A. Perrig, L. van Doorn, and P. Khosla, "SWATT: softWare-based attestation for embedded devices," in *Proceedings of the IEEE Symposium on Security and Privacy (IEEE S and P '04)*, pp. 272–282, May 2004.

[30] R. D. Pietro, L. V. Mancini, C. Soriente, A. Spognardi, and G. Tsudik, "Catch me (If you can): data survival in unattended sensor networks," in *Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '08)*, pp. 185–194, March 2008.

[31] F. Hu and N. K. Sharma, "Security considerations in ad hoc sensor networks," *Ad Hoc Networks*, vol. 3, no. 1, pp. 69–89, 2005.

[32] R. Brooks, P. Y. Govindaraju, M. Pirretti, N. Vijaykrishnan, and M. T. Kandemir, "On the detection of clones in sensor networks using random key predistribution," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 37, no. 6, pp. 1246–1258, 2007.

[33] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 41–47, Washington, DC, USA, November 2002.

[34] K. Xing, X. Cheng, F. Liu, and D. H. C. Du, "Real-time detection of clone attacks in wireless sensor networks," in *Proceedings of the 28th International Conference on Distributed Computing Systems (ICDCS '08)*, pp. 3–10, Beijing, China, July 2008.

[35] K. Xing, X. Cheng, L. Ma, and Q. Liang, "Superimposed code based channel assignment in multi-radio multi-channel wireless mesh networks," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '07)*, pp. 15–26, September 2007.

[36] W. Znaidi, M. Minier, and S. Ubeda, "Hierarchical node replication attacks detection in wireless sensors networks," in *Proceedings of the 20th IEEE Personal, Indoor and Mobile Radio Communications Symposium (PIMRC '09)*, pp. 82–86, Tokyo, Japan, September 2009.

[37] D. Xia and N. Vlajic, "Near-optimal node clustering in wireless sensor networks for environment monitoring," in *Proceedings of the 21st International Conference on Advanced Networking and Applications (AINA '07)*, pp. 632–641, IEEE Computer Society, Washington, DC, USA, 2007.

[38] C. M. Yu, C. S. Lu, and S. Y. Kuo, "CSI: compressed sensing-based clone identification in sensor networks," in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops '12)*, pp. 290–295, Lugano, Switzerland, March 2012.

[39] A. J. Menezes, S. A. Vanstone, and P. C. V. Orschot, *Handbook of Applied Cryptography*, CRC Press, New York, NY, USA, 1996.

[40] C. Bekara and M. Laurent-Maknavicius, "A new protocol for securing wireless sensor networks against nodes replication attacks," in *Proceedings of the 3rd IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '07)*, White Plains, NY, USA, October 2007.

[41] C. Bekara and M. Laurent-Maknavicius, "Defending against nodes replication attacks on wireless sensor networks," 2012, http://www-public.it-sudparis.eu/ lauren_m/articles/bekara-SARSSI07.pdf.

[42] M. Conti, R. Di Pietro, L. V. Mancini, and A. Mei, "A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, pp. 80–89, September 2007.

[43] M. Conti, R. Di Pietro, L. Mancini, and A. Mei, "Distributed detection of clone attacks in wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 5, pp. 685–698, 2011.

[44] B. Zhu, V. G. K. Addada, S. Setia, S. Jajodia, and S. Roy, "Efficient distributed detection of node replication attacks in sensor networks," in *Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC '07)*, pp. 257–266, Miami Beach, Fla, USA, December 2007.

[45] B. Zhu, S. Setia, S. Jajodia, S. Roy, and L. Wang, "Localized multicast: efficient and distributed replica detection in large-scale sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 7, pp. 913–926, 2010.

[46] S. Ratnasamy, B. Karp, L. Yin et al., "GHT: a geographic hash table for data-centric storage," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pp. 78–87, September 2002.

[47] F. Fei, L. Jing, and Y. Xianglan, "Space-time related pairwise key predistribution scheme for wireless seneor networks," in

Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '07), pp. 2692–2696, Shanghai, China, September 2007.

[48] L. C. Ko, H. Y. Chen, and G. R. Lin, "A neighbor-based detection scheme for wireless sensor networks against node replication attacks," in *Proceedings of the International Conference on Ultra Modern Telecommunications and Workshops (ICUMT '09)*, pp. 1–6, St. Petersburg, Russia, October 2009.

[49] J. W. Ho, "Distributed detection of node capture attacks in wireless sensor networks," in *Smart Wireless Sensor Networks*, H. D. Chunch and Y. K. Tan, Eds., pp. 345–360, InTech, Rijeka, Croatia, 2010.

[50] A. Wald, *Sequential Analysis*, Dover, New York, NY, USA, 2004.

[51] C. A. Melchor, B. Ait-Salem, P. Gaborit, and k. Tamine, "Active detection of node replication attacks," *International Journal of Computer Science and Network Security*, vol. 9, no. 2, pp. 13–21, 2009.

[52] Z. Li and G. Gong, "Randomly directed exploration: an efficient node clone detection protocol in wireless sensor networks," in *Proceedings of the 6th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS '09)*, pp. 1030–1035, Macau, China, October 2009.

[53] X. Meng, K. Lin, and K. Li, "Note based randomized and distributed protocol for detecting node replication attack," in *Algorithms and Architectures for Parallel Processing*, vol. 6081 of *Lecture Notes in Computer Science*, pp. 559–570, 2010.

[54] J. W. Ho, D. Liu, M. Wright, and S. K. Das, "Distributed detection of replica node attacks with group deployment knowledge in wireless sensor networks," *Ad Hoc Networks*, vol. 7, no. 8, pp. 1476–1488, 2009.

[55] Y. Sei and S. Honiden, "Distributed detection of node replication attacks resilient to many compromised nodes in wireless sensor networks," in *Proceedings of the 4th Annual International Conference on Wireless Internet (WICON '08)*, 2008.

[56] C. Kim, S. Shin, C. Park, and H. Yoon, "A resilient and efficient replication attack detection scheme for wireless sensor networks," *IEICE Transactions on Information and Systems*, vol. 92, no. 7, pp. 1479–1483, 2009.

[57] B. Dutertre, S. Cheung, and J. Levy, "Lightweight key management in wireless sensor networks by leveraging initial trust," SDL Technical Report SRI-SDL-04-02, 2004.

[58] J. W. Ho, M. Wright, and S. K. Das, "Fast detection of mobile replica node attacks in wireless sensor networks using sequential hypothesis testing," *IEEE Transactions on Mobile Computing*, vol. 10, no. 6, pp. 767–782, 2011.

[59] J. W. Ho, M. Wright, and S. K. Das, "Fast detection of replica node attacks in mobile sensor networks using sequential analysis," in *Proceedings of the IEEE INFOCOM*, pp. 1773–1781, Rio de Janeiro, Brazil, April 2009.

[60] X. M. Deng and Y. Xiong, "A new protocol for the detection of node replication attacks in mobile wireless sensor networks," *Journal of Computer Science and Technology*, vol. 26, no. 4, pp. 732–743, 2011.

[61] C. M. Yu, C. S. Lu, and S. Y. Kuo, "Mobile sensor network resilient against node replication attacks," in *Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '08)*, pp. 597–599, June 2008.

[62] C. M. Yu, C. S. Lu, and S. Y. Kuo, "Efficient and Distributed Detection of Node Replication Attacks in Mobile Sensor Networks," in *Proceedings of the 70th IEEE Vehicular Technology Conference (VTC Fall '09)*, pp. 20–23, Anchorage, Alaska, USA, September 2009.

[63] L. M. Wang and Y. Shi, "Patrol detection for replica attacks on wireless sensor networks," *Sensors*, vol. 11, no. 3, pp. 2496–2504, 2011.

[64] Y. Lou, Y. Zhang, and S. Liu, "Single hop detection of node clone attacks in mobile wireless sensor networks," in *Proceedings of the International Workshop on Information and Electronics Engineering (IWIEE)*, 2012.

[65] W. T. Zhu, J. Zhou, R. Deng, and F. Bao, "Detecting node replication attacks in mobile sensor networks: theory and approaches," *Security and Communication Networks*, vol. 5, no. 5, pp. 496–507, 2012.

[66] M. Conti, R. Di Pietro, L. V. MAncini, and A. Mei, "Emergent properties: detection of the node-capture attack in mobile wireless sensor networks," in *Proceedings of the 1st ACM Conference on Wireless Network Security (WiSec '08)*, pp. 214–219, Alexandria, Va, USA, 2008.

[67] X. Deng, Y. Xiong, and D. Chen, "Mobility-assisted detection of the replication attacks in mobile wireless sensor networks," in *Proceedings of the 6th Annual IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '2010)*, pp. 225–232, October 2010.

[68] I. Amundson and X. D. Koutsoukos, "A survey on localization for mobile wireless sensor networks," in *Proceedings of the 2nd International Conference on Mobile Entity Localization and Tracking in GPS-Less Environments (MELT '09)*, vol. 5801 of *Lecture Notes in Computer Science*, pp. 235–254, 2009.

[69] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the IEEE Symposium on Security And Privacy (IEEE S and P '03)*, pp. 197–213, May 2003.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 407971, 13 pages
http://dx.doi.org/10.1155/2013/407971

Research Article

An Efficient Biometric Authentication Protocol for Wireless Sensor Networks

Ohood Althobaiti, Mznah Al-Rodhaan, and Abdullah Al-Dhelaan

Computer Science Department, King Saud University, Riyadh, Saudi Arabia

Correspondence should be addressed to Mznah Al-Rodhaan; rodhaan@ksu.edu.sa

Received 25 January 2013; Accepted 20 April 2013

Academic Editor: Muhammad Khurram Khan

Wireless sensor networks (WSNs) are spreading rapidly due to their flexibility to communicate which demands a secure environment. The most important requirements of WSN security are confidentiality, authentication, and integrity. User authentication is necessary for legitimate access control in WSNs. Sensors have limited processing power, bandwidth, memory, and limited communication abilities. Significantly, the system must produce an authentication method to confirme if the user is legal or not. In this paper, we present a solution based on biometric and adapt it for a WSN environment. The proposed protocol involves simple operations and light computations. The main advantage of the proposed protocol is using the user's iris to regenerate the user's key on-the-fly every time the user wants be authenticated which dramatically enhances security aspects in WSNs. The key used in this protocol is stronger than passwords and shorter than biometric data, which balances between security and performance. Our protocol uses much light computations and simple operations in both homogenous and heterogeneous environments; therefore, it is suitable to the WSNs.

1. Introduction

Wireless sensor network (WSN) collects the observed data about an environment over a certain geographic area [1]. Users can request and watch the data when they need it (ad hoc queries) or when an event has been triggered.

Very soon, the environmental data will be omnipresent. The reason behind that are the pervasive nature of WSN and its simplicity of deployment which helps in proposing various types of WSN implementations. Such usage of a WSN for smart constructions and the prediction of heat and dampness measurements in a specific position of an area will be on request. Commonly, the great number of requests in WSN implementations is processed at the gateway or the base station points. Nevertheless, we could predict the increasing of higher requests to reach the real-time data from WSNs. This will prevent the real-time data from being reached just at the gateway nodes or at the base stations, rather it could be reached from a sensor node of a WSN in an ad hoc way. Nonetheless, in several situations the use of such gateway node or base station is impossible or not achievable forcing the user to access the sensor devices directly. For instance, data might be located in the directly accessed nodes. In such situation, inquiries and data dissemination have to be performed through the WSN for security purposes. For example, the temporary deployment in battlefield or area where there is no existing network infrastructure is the situation with deployment deep in desert or forest.

The collected data might not be critical (e.g., the humidity in a specific location inside a house), while in some cases they might be confidential and valuable. In the latter case, some security measures must be considered to protect these critical data and not to allow the accessibility of unauthorized users to reach these data. One of the well-known problems in most of the available computer applications and systems is the access control. One of the primary solutions is the user authentication which used to prove the identity of user or a machine as client to access the system or application. User authentication examples can be faced regularly, like entering a local area network of our work place, verification of hand phone appliances, down to the verification of our account transaction of ATM machines, and so on. Nevertheless, a great effort has been done on WSN security

but there are many opened issues that need to be tickled taking into consideration the nature of resource constraints of WSN such as computation limitations, storage, and battery power. It is very challenging to implement the traditional solution of user authentication for wired environments to WSNs.

In this paper, we will discuss the problem of the user authentication in the context of WSNs since a legal user is permitted to query the data in an ad hoc way from any sensor node within WSN and present the needed literature review in Section 2. Furthermore, we suggest a biometric-based user authentication protocol in Section 3. Moreover, Section 4 analyses the security aspects of our protocol. In Section 5, we analyse the protocol performance using analytical modeling. Performance evaluation for the proposed protocol using simulator is presented in Section 6, while Section 7 concludes our study.

2. Literature Review

In a WSN application layer, many security protocols were applied to enhance the security of the layer efficiently, since the standard security protocols in an application layer are considered not strong enough for many information systems privacy [2]. In [3], Benenson et al. propose a user authentication protocol which is used to handle node capture attack. This protocol needs exponential computations because of their protocol based on elliptic curve discrete logarithm problem (ECDLP). As a result, the computational time of Benenson et al.'s protocol is high because exponential computations are expensive. Also, this protocol relies on the existence of a trusted third party.

In 2006, Wong et al. [4] has proposed a dynamic WSN authentication scheme based on a light-weight strong password. This scheme involves three phases: registration phase, login phase, and authentication phase. However, Wong et al.'s scheme has three main advantages that are discussed as follows.

(i) It gives the authorized users privilege to access data at any sensor nodes in an ad hoc method.

(ii) It reduces the computation cost by loading very little operations.

(iii) It is secure enough against some of replay/forgery attacks.

Whereby, this scheme has four disadvantages.

(i) It has some weakness in protecting against all replay/forgery attacks.

(ii) Sometimes sensor node reveals and exposes the password information to other nodes.

(iii) It does not allow users to change their password freely.

(iv) The scheme is always vulnerable to stolen-verifier threats.

Tseng et al. [5] have improved Wong et al.'s scheme by fixing the security weakness in it. This improvement has been designed by adding an extra phase on Wong et al. phases. However, Tseng et al.'s scheme has four phases: registration, login, authentication, and password changing phase.

Novelty, Ko [6] showed that the authentication process shown in [5] is still insecure and does not achieve mutual authentication, which is very important for many applications. Moreover, Vaidya et al. [7] highlighted other weaknesses on Tseng et al., Wong et al., and Ko schemes. Vaidya et al. pointed out that the previous mentioned schemes are still not strong enough to protect against some attacks such as replay of account login, stolen verifier, and man-in-the-middle attacks. Vaidya et al. enhanced the security by proposing two WSN authentication schemes depending on the traditional password authentication schemes.

However, Das [8] has also highlighted the weakness of Wong et al. against stolen-verifier attacks, as well as identified two-factor user authentication scheme for WSN which is based on password and smart card. Although this two-factor scheme can be used to protect networks but it requires some complicated equipment.

In [9], He et al. have developed and discussed two-factor authentication scheme. This development is proposed to enhance the security of the two-factor scheme by dividing their scheme into three main phases: registration, authentication, and password-changing phase.

However, He et al. have developed another issue from Das's scheme. This developed issue is still suffering from some security limitations, such as session key establishment (where the session key between user and node sensor is not established after authentication phase in the developed scheme) and authentication process between user and sensor node.

Consequently, Nyang and Lee [10] showed that Das's scheme is still vulnerable and has some security weakness in offline-password-guessing attacks. As a result, Das's scheme cannot control and protect the established query-response messages between sensor node and user. Therefore, Nyang and Lee proposed a scheme to overcome the important security weakness in Das's scheme.

Khan and Alghathbar [11] studied Das's scheme in more details. They mentioned that Das's scheme is still vulnerable to several types of attack, since Das's scheme does not provide some features such as password changing and gateway and sensor mutual authentication. Therefore, their suggestions to fix this weakness are by adding password changing phase and providing gateway/sensor mutual authentication phase. The suggested scheme is still weak because of the limitations on the session key establishment between the user and sensor node in WSN.

Arikumar and Thirumoorthy [12] proposed their contribution in securing WSN depending on the two-factor authentication scheme and nonpublic key operations such as hash function. This contributed scheme prevents users who have the same identity from logging into the system.

Yeh et al. [13] presented a new user authentication scheme depending on elliptic curve cryptosystem and smart card authentication method. This contribution is still weak in the following regards.

(i) Communication cost is very expensive compared to the existing schemes.

(ii) Working without user/sensor node mutual authentication.

(iii) Working without key sharing between the user and the sensor node.

Recently, Yuan et al. [14] used biometric approach to propose their user authentication scheme in WSN. In addition, they used password and smart card in their presented work. After Yuan et al.'s biometric scheme, Yoon and Yoo [15] proposed a new user authentication protocol based on biometric approach and without using password. They also showed the message integrity problem in Yuan et al.'s scheme. Although they suggested security improvements of Yuan et al.'s protocol, but some drawbacks are still in their protocol, for example, there is no session key established after authentication between the sensor node and the user, and no confidentiality of messages is considered; also their protocol requires some complicated equipment and is still vulnerable to several types of attack such as denial of service (DoS) attack.

3. Proposed Protocol

We observed that the existing protocols have security weaknesses, such as, DoS, man-in-the-middle attack (MIMA), and guessing attack. No confidentiality of messages is considered and may require some complicated equipment. We suggest security improvements in our protocol to fix some of the drawbacks to improve the existing user authentication in WSNs. Our protocol not only solves the aforementioned drawbacks but also improves the security of user authentication in WSNs.

Our user authentication protocol is based on biometric encryption and hash function which is feasible for WSNs without special hardware support (i.e., without additional equipment) and without third party. Moreover, our protocol is suitable for large scale applications.

The main advantage which dramatically enhances security aspects in WSNs is that the user's iris is used to regenerate the user's key on-the-fly every time the user wants to be authenticated. This encryption key is shorter than biometric data and stronger than a password, that balances the tradeoff between performance and required security; it becomes possible to achieve biometric-based user authentication without transmitting and saving any private information anywhere (i.e., no need to store neither images nor the template of them in the memory. The encryption key and the image or template of image must be discarded at the end of the registration phase).

Our major goal is to decrease possible problems which are caused by illegal users. So we suggest a user authentication protocol to fulfill the following requirements.

(i) Our protocol provides protection against main attacks in WSNs such as replay attack, impersonation attack, MIMA, stolen-verifier attack, repudiation attack, data corruption attack, password guessing attack, and DoS attack. The security of conventional user authentication protocols is founded on password. Short password is broken without difficulty using password guessing attacks. Furthermore, a password can be shared with other people or be lost so there is no method to know who is the real user. Biometric encryption can solve the above security problems, that is, based on behavioral or physiological characteristics of persons, for example, fingerprints, iris scan, faces, hand geometry, vein patterns, voice patterns, and so forth. The biometric-based authentication is more reliable than conventional authentication based on a password.

(ii) Our protocol provides mutual authentication between not only gateway and sensor node but also between gateway and user.

(iii) Our protocol was adapted to be efficient and lightweight in terms of computational cost and communication cost to decrease the energy consumption of sensor nodes which have limited energy and resources.

(iv) Our protocol is based on a zero-knowledge proof. This means that it allows a claimant to prove the knowledge of a secret without revealing it.

(v) Our protocol provides confidentiality of messages between all entities (user, gateway, and sensor node); therefore, only authorized users can use these messages, which are confidential against any attack.

The proposed protocol uses iris encryption in user authentication for wireless sensor networks. In this protocol, there are four main phases: registration, login, authentication, and user's key change. We chose iris scan as the most appropriate personnel trait for authenticating users, because it was demonstrated that iris is one of the most accurate traits for user authentication. Other advantages for authentication based on iris scan are found in the literature [16]. Moreover, iris authentication may need no additional hardware on the user devices with the presence of digital cameras now are included in most computer devices [17]. The number of commercial software produces iris recognition using mobile devices cameras, for example, BioWallet on Android and OKI's iris recognition on Symbian.

We have made the following assumptions for user authentication.

(i) The gateway node (GW-node) or base station (BS) is considered as trusted node.

(ii) Each user must register in the system once to access the WSN data.

The notations used in the protocol are explained in Table 1.

After extracting the iris's features, the biometric encryption will take place by using a fuzzy commitment scheme, as in [17]. Fuzzy commitment scheme overcomes the drawback of traditional biometric systems where there is no need to store neither images nor the template of them in the memory.

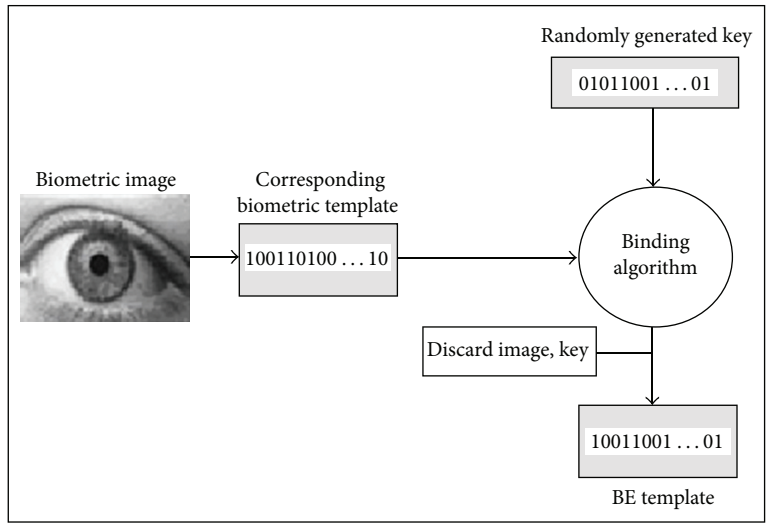


Figure 1: Biometric cryptography.

Table 1: Notations used in the protocol.

Notation	Meaning
U	User
SN	Identity of sensor login-node N, that is, nearest sensor node of WSN
ID	User's identity
$h(\cdot)$	One-way Hash function
$\|$	Concatenation
$E_{\text{key}}(m)$	Message m is encrypted with secret key
$D_{\text{key}}(m)$	Message m is decrypted with secret key
$\text{MAC}_{\text{key}}(m)$	Message authentication code

Biometric data has a variable nature and the encryption needs an exact key to work well; so before biometric data can be used as a key for encryption, its representation must be stabilized. Error-correction codes are used for the stabilization process [17].

Registration phase of our protocol is performed by extracting the features of iris using an iris recognition system, then binding (XORing) the corrected biometric data with a random generated key (user's key). Then the biometric encryption (BE) template will be saved in the user's device (PDA/PC).

Additionally, the hash value of encryption key will be saved with the BE template to be able to reject incorrect keys in an early step, before beginning the process of remote authentication as shown in Figure 1.

When the BE template is saved in the user's device, the user can retrieve his key by capturing an image of his iris via the user's device camera. After that, the features of iris will be extracted by the iris recognition model, then XORed with the BE template to regenerate the user's key as shown in Figure 2.

There are two methods to authenticate the user using the biometric encryption. One of them uses the biometric to create a pair of private and public key, as proposed by [16], and publish the public key. The other method uses biometric encryption to generate a user's secret key [18]. The problem in the second method is the decision of what should be saved on the GW-node/BS to authenticate the user. The first answer is the "BE template," as it is in a normal biometric authentication. Unluckily, this is not useful from the viewpoint of the biometric, because the user must transmit his/her biometric data to the remote authenticator for retrieving his/her key from the GW-node database. The best solution is that the key of user is saved on the GW node. Although this solution contradicts with the biometric encryption aim to not save the key of user, but this inconsistency will be on the GW-node side only which has a high level of security, particularly if the users' keys were encrypted in the GW-node database. Hence, the security of the GW-node is not the matter because this method is similar to the well-known Kerberos protocol. Furthermore, if the user's key is stolen, it can be simply revoked using the reenrollment process. We use the second method for our user authentication protocol because the symmetric cryptographic is less complex than asymmetric cryptography and consumes less resources, which is an important issue in limited resources devices, for example, sensor node and mobile devices.

3.1. Registration Phase. When the user U_i registers in the system, an encryption key will be generated randomly for U_i. The generated key is saved on the GW node as a key of U_i. After that, the features of the U_i's iris are extracted then hashed by SHA256 [17], then the hash is XORed with the key to generate a BE template which will be saved in the U_i's device to use it for the authentication phase to regenerate the key from the U_i's iris. The template of iris is large, also if we use a part of the iris template in the biometric encryption that is not secured enough, so the hash was used because a fixed size output for variable length inputs is produced by the hash function and the hash function is collision resistant. Additionally, the hash of the key will be saved with the BE template to be able to reject incorrect keys in an early step,

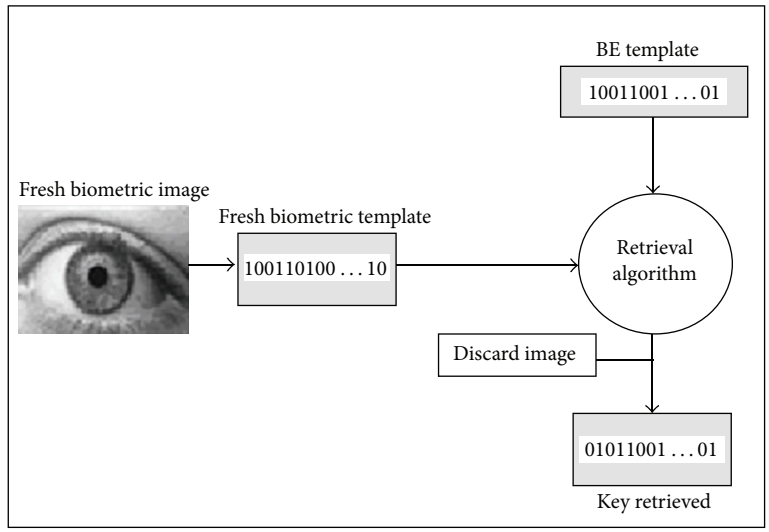


Figure 2: Decrypt with same biometric.

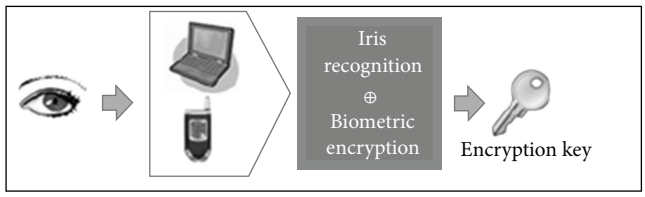


Figure 3: Login phase.

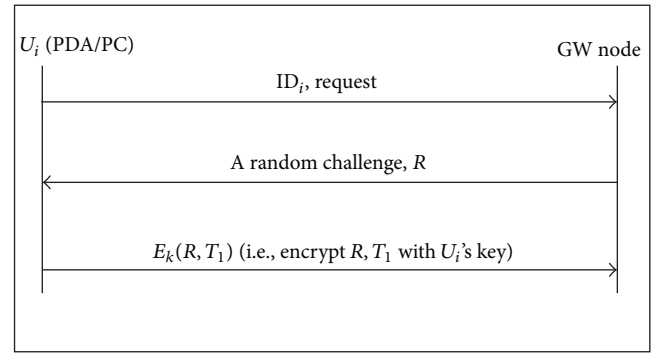


Figure 4: Authentication between user U_i and GW node.

before beginning the procedure of remote authentication. The U_i data (ID_i, name, etc.) and the encryption key of the U_i are saved in the GW-node database. After that, the GW node sends to the U_i an ID_i and $F_i = h(ID_i \oplus X)$ through a secure channel to utilize them at an authentication phase to authenticate himself. Where X is a secret parameter which is generated by the GW-node and it is saved in all the SNs (the sensor login-nodes) before the nodes are deployed in the field. These sensor nodes (SNs) will be responsible to respond to the data/query that users are looking for and know X.

3.2. Login Phase. After iris acquisition by camera in the U_i's device, the features of U_i's iris are extracted. After that, the iris's features are corrected by error correcting code and hashed by SHA256. Then, the hash of corrected iris's features is XORed with the saved BE template to regenerate the U_i's key as in Figure 3. Then, the U_i's key is hashed, and this hash will be compared with the saved hash of the U_i's key. If the two hashes are not equal, the remote authentication is aborted. If they are equal, the application is proceeded. After that, a request is sent to the GW node including the user ID_i.

3.3. Authentication Phase. When GW node receives the login request, it replies with a random challenge R. Here the GW node would challenge the user to encrypt a bit of known information using the encryption key. When the user receives the GW-node response, he/she will encrypt R and T_1, where T_1 denotes the current timestamp of the U_i's device, using the encryption key which is generated from the iris template and sent the encrypted message to GW node. When GW node receives encrypted message at time T_2, it will decrypt this message using the key of U_i then checks the freshness of timestamp T_1 as in Figure 4. If $(T_2 - T_1) > \Delta T$, the authentication phase will be aborted, where ΔT denotes interval of the expected time for the transmission delay in the WSN. In contrast, if $(T_2 - T_1) \leq \Delta T$, the following steps will be achieved.

The next step is that GW node computes $F_i = h(ID_i \oplus X)$, then computes $Y_i = MAC_{F_i}(ID_i \parallel SN \parallel T_3)$, where SN denotes the sensor node which will reply to the query with what U_i is looking for and T_3 denotes the GW node's current timestamp. The GW node transmits a message (ID_i, Y_i, T_3) to the SN over a public channel. Y_i used by SN to ensure that the message (ID_i, Y_i, T_3) issues from the legal GW node since Y_i is produced using X, that is known to the GW node and SN.

When the request is received by the SN at time T_4, it will carry out the following steps, the verification of T_3. If $(T_4 - T_3) > \Delta T$, the authentication phase will be aborted, where

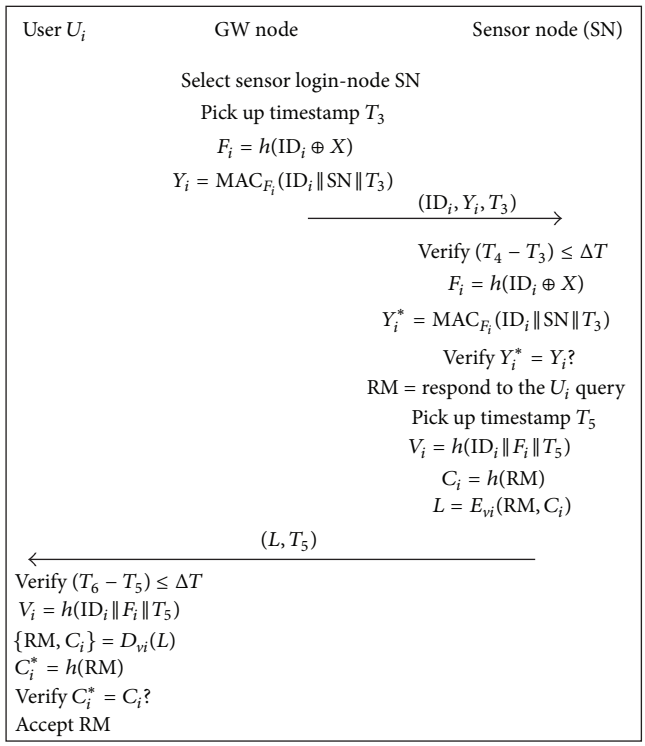


Figure 5: Authentication phase.

ΔT denotes interval of the expected time for the transmission delay in the WSN. In contrast, if $(T_4 - T_3) \le \Delta T$, the following step will be achieved. The SN computes $F_i = h(\mathrm{ID}_i \oplus X)$ and $Y_i^* = \mathrm{MAC}_{F_i}(\mathrm{ID}_i \| \mathrm{SN} \| T_3)$; after that, it verifies whether or not Y_i^* is equal to Y_i. If the previous two verifications successfully pass, the SN sets respond to the U_i query (RM), then compute $V_i = h(\mathrm{ID}_i \| F_i \| T_5)$ and $C_i = h(\mathrm{RM})$, where T_5 denotes the sensor node's current timestamp, then compute $L = E_{vi}(\mathrm{RM}, C_i)$ (i.e., SN encrypts RM and C_i with V_i); after that, the SN sends a message (L, T_5) to the U_i.

In next step, when the U_i receives the message (L, T_5) at time T_6, U_i verifies from freshness of timestamp T_5. If $(T_6 - T_5) > \Delta T$, the authentication phase will be aborted. In contrast, if $(T_6 - T_5) \le \Delta T$, U_i computes $V_i = h(\mathrm{ID}_i \| F_i \| T_5)$ and then decrypts the message L with V_i (i.e., U_i computes $D_{vi}(L)$) to get RM and C_i and then computes $C_i^* = h(\mathrm{RM})$; after that, he/she verifies whether or not C_i^* is equal to C_i. If the previous steps successfully pass, the U_i will accept the RM as explained in Figure 5. Furthermore, the SN and U_i have become the shared session key $V_i = h(\mathrm{ID}_i \| F_i \| T_5)$; this session key can be used to achieve more operations during a session. Therefore, a legal user communicates to sensor nodes in an ad hoc way to access data of the WSN.

3.4. User's Key Change Phase. According to the proposed protocol, the user's key can be changed by re-enrollment process. When the key is compromised, U_i can do re-enrollment by his/her biometric, and then a new key is generated randomly; this key differs from the previous.

4. Security Analysis Using Threat Model

This section proves our protocol's strength in terms of security. We demonstrate that the proposed protocol resists main types of attacks in WSNs which are found in the literature.

In communication networks, the threat model is employed to analyze crypto protocols formally, since the threat model assumes that two parties can communicate over an insecure channel. WSNs can adopt the threat model where the channel of communication between two parties is insecure, and the end points (sensor node and user) cannot be trusted generally.

Proclamation 1. The proposed protocol resists a DoS attack.

Proof. The DoS attack in the existing user authentication protocols can be occurred by attacker who is transmitting the large number of requests to GW node in login and authentication phases to make the GW node fail. Our protocol prevents DoS attack by preauthentication, since each request should associate with timestamp T_1 encrypted by the U_i's key; for this reason, a large number of unauthorized requests cannot get into the GW node.

$U_i \rightarrow$ GW-node: preauthentication $= E_K(R, T_1)$, where T_1 denotes the current timestamp of the U_i's device. □

Proclamation 2. The proposed protocol provides confidentiality of messages between all entities (user, gateway, and sensor node).

Proof. Messages confidentiality against eavesdropping attack are performed by data encryption service. Our protocol can provide sufficient confidentiality for transmitted messages (e.g., $E_K(R, T_1)$ and $L = E_{vi}(\mathrm{RM}, C_i)$). More specifically, these messages are confidential against any adversary. If data is sent without encryption over a public channel, the attacker is able to view the plaintext data as it passes over the network; this attack occurs in Yoon and Yoo's protocol [15], where in [15] the sensor node's responding message (RM) is sent to the user over a public channel without encryption. □

Proclamation 3. The proposed protocol resists a node compromise attack.

Proof. WSNs are normally deployed in an unattended environment. The attacker can easily capture a sensor node (SN) and attempt to collect some secret information about the network from this sensor node. Implementation of one-time sensors prevents this type of the attack, but it is limited to some applications (e.g., fire alarm), because the data confidentiality is not important. When the data confidentiality is important, it is a difficult task to prevent this type of the attack if sensor nodes are not tamper-proof and the environment is hostile and unattended. However, the GW node can periodically monitor whether any sensor node is captured or not. If authentication of user and data access from sensor node are allowed directly to the user (i.e., without GW node's notice) then the effect of "node compromise" attack

is very high, which occurs in Watro et al.'s scheme [19]. But in our protocol, the user's request is first authenticated by the GW node and after that the request is transmitted to the sensor node to respond to the user query. □

Proclamation 4. The proposed protocol resists a replay attack.

Proof. As a rule, if previously obtained information is not reusable, then replay attacks are impossible [20]. When an opponent eavesdrops on the communication between the U_i and the GW node, he/she only gets encrypted data (unreadable form), which is not reusable. Therefore, no opponents's success replay attacks on the proposed protocol.

The timestamps T_i $(i = 1, 2, \ldots, 6)$ are used in the suggested protocol to prevent the replay attack. If an adversary intercepts the message $E_K(R, T_1)$ and attempts replaying the same message for login to the GW node, he/she cannot pass the verification of the login request because of $(T_2 - T_1) > \Delta T$, where T_2 denotes the time when the replayed message is received by the GW node. Similarly if an adversary intercepts $(\mathrm{ID}_i, Y_i, T_3)$ and attempts replaying the same message, he/she cannot pass the verification of the login request because of $(T_4 - T_3) > \Delta T$. Also if an adversary intercepts (L, T_5) and attempts replaying the same message, he/she cannot pass the verification of the login request because of $(T_6 - T_5) > \Delta T$. □

Proclamation 5. The proposed protocol resists an impersonation attack.

Proof. The suggested protocol resists impersonation attack since an adversary may intercept a login request $E_K(R, T_1)$, nevertheless, to log in again, $E_K(R, T_1)$ must be recomputed by using a new timestamp T_{new}, for avoiding the replay attacks. This is impossible without U_i's iris since U_i's iris is required to regenerate the U_i's key, and the user's biometrics cannot be attained by the adversary. Therefore, the adversary cannot achieve impersonate attack. Therefore, the proposed scheme can resist user impersonation attack. Also the adversary cannot impersonate a valid GW node. Assume that an adversary intercepts a valid login message $(\mathrm{ID}_i, Y_i, T_3)$ to impersonate a valid GW-node. Nevertheless; because the MAC function must be one way where MAC function must be hard to invert (i.e., given random $y \in \{0, 1\}^n$, hard to find any x such that $h(x) = y$), the adversary cannot extract secret value $F_i = h(\mathrm{ID}_i \oplus X)$ from $Y_i = \mathrm{MAC}_{F_i}(\mathrm{ID}_i \parallel \mathrm{SN} \parallel T_3)$. Without knowing F_i, the adversary cannot perform the GW-node impersonation attack since the adversary cannot make a new message $(\mathrm{ID}_i, Y_i^{\mathrm{new}}, T_{\mathrm{new}})$, where $Y_i^{\mathrm{new}} = \mathrm{MAC}_{F_i}(\mathrm{ID}_i \parallel \mathrm{SN} \parallel T_{\mathrm{new}})$ and T_{new} denotes a timestamp of the adversary. Consequently, the suggested protocol resists the GW-node impersonation attack. For the same reason, the adversary cannot impersonate a valid sensor node (SN) without knowing the secret value F_i since he/she cannot obtain F_i from the intercepted value $L = E_{vi}(\mathrm{RM}, C_i)$ where $V_i = h(\mathrm{ID}_i \parallel F_i \parallel T_5)$. □

Proclamation 6. The proposed protocol resists a stolen verifier attack.

Proof. The attackers who have stolen the users' keys from GW node cannot obtain any useful information, because the users' keys were encrypted in the GW-node database. □

Proclamation 7. The proposed protocol resists guessing attacks.

Proof. This type of attack is a serious concern in systems based on password. The suggested protocol resists password guessing attacks, because the password of user is not required to login to the WSN. Also the suggested protocol resists the biometric guessing attacks, because our protocol does not require transmitting and saving any private information anywhere. In other words, no preservation of the biometric template or image is done. The best practice from a privacy point of view, in the first place, is not to collect any personally identifiable information (PII) to a completely and fully possible capacity. This is known as "data minimization" which means to minimize the quantity of personal data that are possessed and collected; in this manner, it eradicates the likely preceding abuse. The retention and wrong usage of biometric data are considered as the main reasons of driving most concerns of privacy and security. These threats and concerns are addressed by biometric encryption. □

Proclamation 8. The proposed protocol resists a repudiation attack.

Proof. This type of attack indicates to participation denial in all of the communication or part of it. Our protocol requires U_i's iris to regenerate the U_i's key; therefore, the U_i cannot deny that he/she performed a specific participation; also we assume the GW node is considered as trusted node; thus, our protocol resists a repudiation attacks. □

Proclamation 9. The proposed protocol can resist an integrity threat.

Proof. Data integrity threats faced by our protocol are as follows:

(i) data modification attack,

(ii) data corruption attack,

(iii) data insertion attack.

Integrity is a service used to guarantee that the transmitted data has not been modified via an unauthorized entity. Our protocol can resist integrity threats, since $Y_i = \mathrm{MAC}_{F_i}(\mathrm{ID}_i \parallel \mathrm{SN} \parallel T_3)$ and $C_i = h(\mathrm{RM})$ could be used to protect against integrity threats. The sensor node (SN) can guarantee that the message $(\mathrm{ID}_i, Y_i, T_3)$ has not been modified via an unauthorized entity by recomputing MAC and verifies whether it is equal to the MAC attached to the message. Also the U_i can guarantee that the message (L, T_5) has not been modified via an unauthorized entity by recomputing $h(\mathrm{RM})$ and verifies whether it is equal to the $h(\mathrm{RM})$ attached to the message. □

Proclamation 10. The proposed protocol resists an insider attack.

Table 2: Comparison among representative protocols and our protocol.

	Benenson et al.'s protocol	Yuan et al.'s protocol	Yoon et al.'s protocol	Our protocol
Stolen-verifier attacks	Secure	Secure	Secure	Secure
Guessing attacks	Secure	Secure	Secure	Secure
Impersonation attacks	Insecure	Insecure	Secure	Secure
Replay attacks	Secure	Secure	Secure	Secure
Insider attack	Secure	Insecure	Secure	Secure
Repudiation attack	Secure	Secure	Secure	Secure
Node compromise attack	Secure	Insecure	Insecure	Secure
Denial-of-service attack	Insecure	Insecure	Insecure	Secure
Message confidentiality	Not provided	Not provided	Not provided	Provided
Password change	Not required	Required	Not required	Not required
Key revocability	Not provided	Not provided	Not provided	Provided
Complicated equipment	Not required	Required	Required	Not required
Session key establishment	Not provided	Not provided	Not provided	Provided
Mutual authentication	Not provided	Not provided	Provided	Provided
Data integrity	Not provided	Not provided	Provided	Provided

Proof. An insider attack is intentionally misused by authorized parties. Our protocol aims to achieve biometric-based user authentication without transmitting and saving any private information anywhere (i.e., no need to store the template of image or image in the memory). The encryption key and the image or template of image must be discarded at the end of the registration phase, since only the BE template will be saved in the U_i's device to use it for the authentication phase to regenerate the key from the U_i's iris. For this reason, any insider attack cannot get the correct biometric and cannot get the correct user's key from the BE template because of the advantages of biometric encryption and the one-way hash function. Also the users' keys were encrypted in the GW-node database.

Table 2 shows the functionality comparisons between our protocol and related protocols. According to Table 2, our protocol not only presents confidentiality of messages but also performs all security requirements and without complicated equipment. From the above descriptions, we conclude that our protocol is more practical than the related protocols. □

5. Analytical-Based Performance Evaluation

In this section, we analyze the performance of our protocol based on a mathematical model and compare with the ones in the literature. We employ the computational overhead (denoted by T) to study the performance. The computation time required by each security primitives as stated by practical implementations on Mica2 motes [21] is recorded in Table 3. The total computational cost of our protocol, Benenson et al.'s protocol, Yuan et al.'s protocol, and Yoon et al.'s protocol is shown in Table 4. We calculate the computational time for registration phase separately because the computational time for registration phase is a one-time task at some period. From Table 4, Benenson et al.'s protocol needs exponential computations because their protocol-based on ECDLP. As a result, the computational time of Benenson et al.'s protocol is the highest among the four protocols because exponential computations are expensive.

Table 3: Execution times on Mica2.

Notation	Description	Time (ms)
T_H	Time for performing one-way hash function (SHA-1)	3.636
T_{MAC}	Time for performing MAC function (HMAC-SHA1)	3.12
T_{RC5}	Time to encrypt/decrypt using RC5	0.26

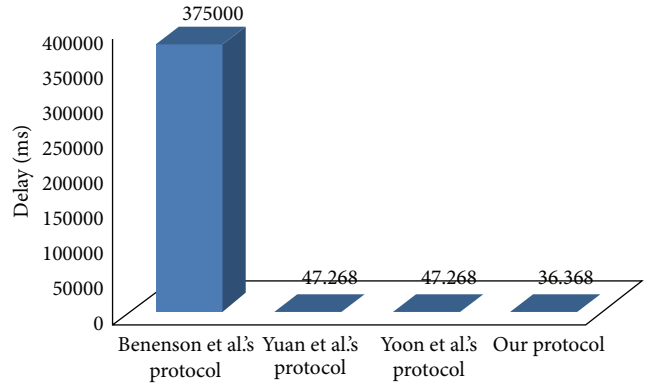


Figure 6: Delay performance.

Also, Benenson et al.'s protocol relies on the existence of a trusted third party while we only assume that the GW node is trusted. Briefly, we minimize the computational time and avoid the want for a trusted third party. Based on Table 4, our protocol requires only 36.368 ms, which is less than Benenson et al.'s protocol (375000 ms), Yuan et al.'s protocol (47.268 ms), and Yoon et al.'s protocol (47.268 ms) as explained in Figure 6.

Table 4: Computational time comparison.

	Benenson et al.'s protocol	Yuan et al.'s protocol	Yoon et al.'s protocol	Our protocol
Registration phase	$1T_{EXP}$	$4T_H$	$3T_H$	$2T_H$
Login and authentication phases	$2nT_H + 3nT_{EXP}$	$9T_H$	$10T_H$	$4T_{RC5} + 2T_{MAC} + 6T_H$
Total	$2nT_H + 3nT_{EXP} + 1T_{EXP}$	$13T_H$	$13T_H$	$4T_{RC5} + 2T_{MAC} + 8T_H$
Total time	375000 ms*	47.268 ms	47.268 ms	36.368 ms

*Time required for the authentication phase on sensor node only for one session [3].
T_{EXP}: the time required to achieve a modular exponential computation.
n: number of sensors in the communication range of the user.

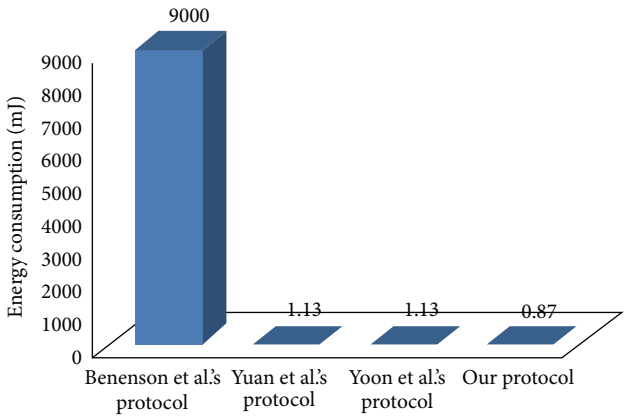


Figure 7: Comparison of energy consumption.

Figure 8: Iris recognition system.

We compute the energy consumption of security computations by (1) to (3) [21]. For Mica2 mote, $I = 8$ mA if the processor status is in active mode. In general, if two new AA batteries are used, $V = 3.0$ V [21]. Therefore, total energy consumption of our protocol is only 0.87 mJ, which is more efficient than Benenson et al.'s protocol (9000 mJ), Yuan et al.'s protocol (1.13 mJ), and Yoon et al.'s protocol (1.13 mJ) as shown in Figure 7:

$$E = V \times Q. \tag{1}$$

Since

$$Q = I \times t, \tag{2}$$

and therefore,

$$E = V \times I \times t, \tag{3}$$

where E denotes the energy consumption, Q is the charge, V is the voltage, I is the current, and t is the elapsed time.

6. Implementation-Based Performance Evaluation

In this section, we introduce the performance evaluation based on the implementation.

6.1. Iris Recognition System. The biometric systems give automatic identifications of the persons derived from a unique characteristics or features. Iris recognition has many advantages such as it has an internal organ (body part), reliable, and does not need to be identified by touching used equipment so iris is considered one of the most important biometric recognition systems.

Our iris recognition system (Figure 8) is performed by Masek's method [22]; some modifications are introduced to Masek's method to achieve the biometric encryption. This iris recognition system uses the global transform to segment the image. Therefore, it can detect the pupil and the iris region. After normalizing the image, the data is extracted and quantized to encode the unique pattern of the iris image using 1D Log-Gabor filters. All tests are performed with CASIA iris image database [23]. Our iris recognition system requires only 4.6570 seconds to extract the features of iris and create BE *template*. For regenerating the user's key, our system requires 4.8280 seconds.

6.2. Simulation-Based Performance Evaluation. In this section, we evaluate the performance of our protocol by MATLAB simulator version R2010a [24]. Results of our simulation comprise not security computational cost only, but also communication cost for transmitting security messages. We will assess the performance of our protocol in the presence of homogeneity and heterogeneity to prove the robustness of

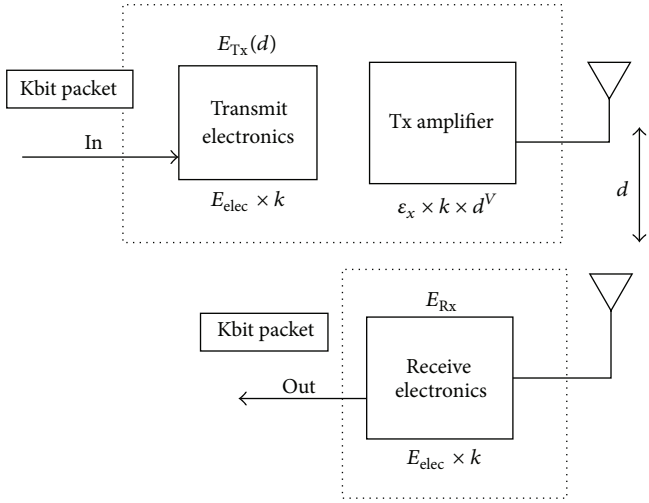


Figure 9: Energy dissipation model.

our protocol, and it works well regardless of network kind. For the aim of this research, we apply similar energy model as discussed in [25, 26] shown in Figure 9.

In this energy model, the energy dissipated per bit (E_{elec}) is 50 nJ/bit for running the receiver or transmitter circuits and the transmit amplifier (ε_{amp}) is 10 pJ/bit/m^2. The energy expended to send k-bits is

$$E_{Tx}(k,d) = E_{Tx\text{-}elec}(k) + E_{Tx\text{-}amp}(k,d),$$
$$E_{Tx}(k,d) = E_{elec} \times k + \varepsilon_{amp} \times k \times d^2, \quad (4)$$

where d is the distance between the sender and the receiver, and E_{Tx} includes the loss of energy because of channel attenuation.

The energy expended to receive k-bits is

$$E_{Rx}(k) = E_{elec} \times k. \quad (5)$$

Also, we assume that the same amount of energy is needed to send k-bits from A to B and vice versa. The parameters used in our simulation are summarized in Table 5.

Security Primitives Choice. Traditional wisdom states select either AES or DES when a block cipher is required. Nevertheless, DES is too slow to be implemented on limited resources devices, and AES is also quite slow. Moreover, AES has the drawback that its block length is long. Skipjack and RC5 are the most appropriate to be implemented on limited resources devices [21]. RC5 has been proven to be secure in the literature [27]. Therefore, we choose RC5 (size of key = 20 byte) as a block cipher. Also, we choose HMAC-SHA1 (size of key = 20 byte) as a MAC function. Value of SHA-1 is 20 bytes.

6.2.1. Homogeneous Wireless Sensor Networks. In this study, we assume that the homogeneous wireless sensor network shown in Figure 10 has the following properties.

(i) Supposing a 120 m × 120 m region of 100 sensors.

Table 5: The parameters used in our simulation.

Parameter	Value
Number of sensor nodes	$n = 100$
Packet size	$k = 4000$ bits
Area	$A = M \times M = 120 \times 120$
GW-node location	(50, 50)
Number of runs	10
MAC protocol	CDMA and TDMA
Communication model	Bi-direction
Transmitter/receiver electronics	$E_{elec} = 50$ nJ/bit
Initial energy for normal node	$E_o = 0.5$ J
Data aggregation energy	$E_{DA} = 5$ nJ/bit/message
Transmit amplifier	$\varepsilon_{amp} = 10$ pJ/bit/m^2

(ii) The sensor nodes are uniformly distributed (i.e., they are randomly distributed in a 2-dimensional space).

(iii) The sensor nodes are static.

(iv) It is supposed that the GW node is placed at the center of the sensing region.

(v) At the network layer, the LEACH (low energy adaptive clustering hierarchy) [25] is used for the routing protocol. Typically, the stability period for LEACH protocol is 995 rounds [28].

The results are present in Figure 11, which compare delay of cases without (w/o) and with (w/) in the proposed protocol and are evaluated during the stability period of network. In the first case (i.e., without the proposed protocol), there were only plain texts transmitted between the accessed node and the user. For the second case (i.e., with the proposed protocol), all security communications and computations of proposed protocol were considered. The communicational and computational cost with the proposed protocol is 99.89 seconds, while without the proposed protocol is 97.37 seconds. This means that the proposed protocol only increases only by 2.59% delay. Briefly, the final results of the simulation demonstrate a small increase of cost of proposed protocol compared with a former case. As a result, such delay is insignificant for the WSN.

Figure 12 shows one that authentication of Benenson et al.'s protocol takes approximately 440 seconds, while one authentication of our protocol takes only 0.53 second. In addition, Benenson et al.'s protocol needs the certificate and public key of a user at the side of receiver, which have been transmitted with each request of user (i.e., rising overhead of transmission). It is clear that the performance of our protocol is high enough compared to that of Benenson et al.'s protocol (by 83574.05%).

6.2.2. Heterogeneous Wireless Sensor Networks. We suppose that heterogeneous wireless sensor network such as Figure 13 has the following properties of heterogeneity.

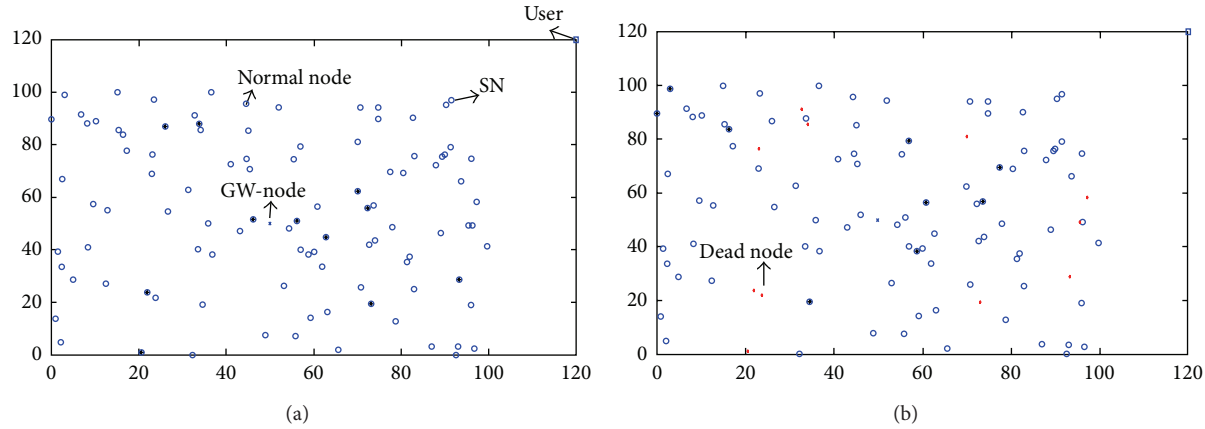


Figure 10: (a) A homogeneous wireless sensor network when all the sensor nodes are alive, (b) the network when some sensor nodes are dead.

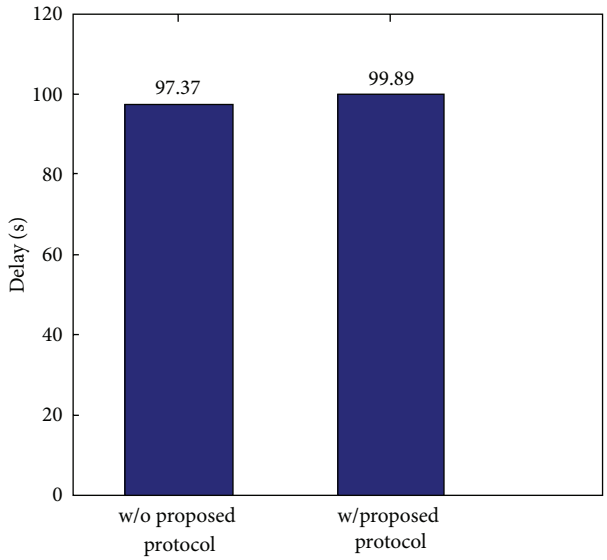


Figure 11: Delay performance of homogeneous network.

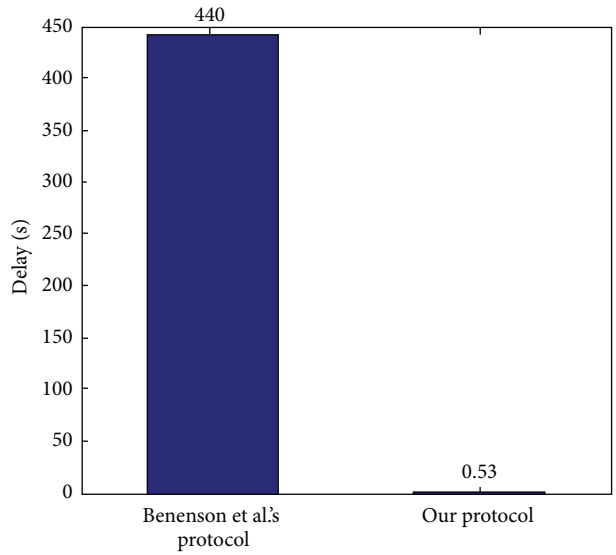


Figure 12: Comparison of delay.

(i) The SEP (stable election protocol) [26] is used at the network layer. Typically, the stability period for SEP protocol is 1385 rounds [28].

(ii) The heterogeneity parameters are the fraction of advanced nodes (m = 10%) and the additional energy level between advanced node and normal node (α = 1).

The communicational and computational cost during the stability period of heterogeneous network with the proposed protocol is 137.89 seconds, while without the proposed protocol is 135.24 seconds as explained in Figure 14. This means that the proposed protocol increases only by 1.96% delay. In brief, the final results of the simulation show a small increase of cost of the proposed protocol compared to a normal case. Therefore, our protocol is robustness and works well regardless of network kind.

7. Conclusion and Future Work

In this paper, we have conducted a study on the security of WSNs in the field of authentication. We have studied the existing protocols of user authentication in WSNs. Furthermore, we suggest a biometric-based user authentication protocol. The above analysis proves that our protocol is more practical than the representative protocols and proves the robustness of our protocol since it works well regardless of network kind.

The main advantage of the proposed protocol, which dramatically enhances security aspects in WSNs, is that the user's iris is used to regenerate the user's key on-the-fly every time the user wants be authenticated. In the future, a hardware implementation (with real sensors) of our protocol will be conducted to analyze its real performance and efficiency. Also, we will improve the iris recognition system to introduce

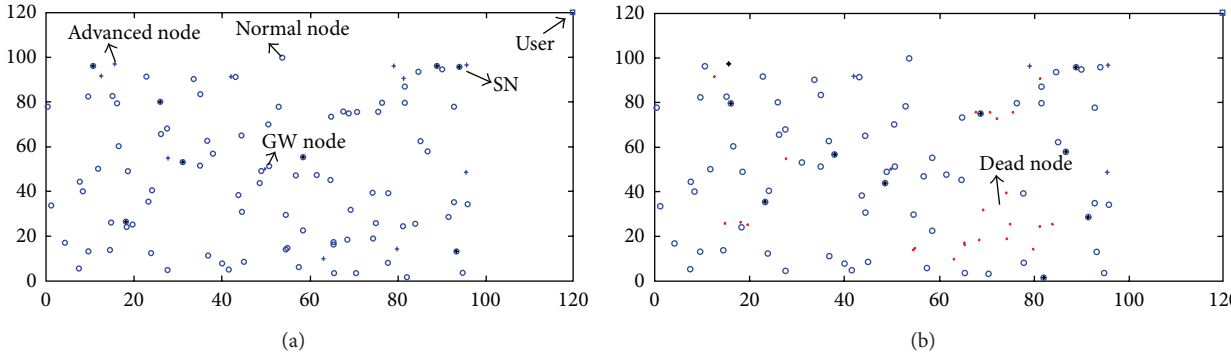


Figure 13: (a) A heterogeneous wireless sensor network when all the sensor nodes are alive, (b) the network when some sensor nodes are dead.

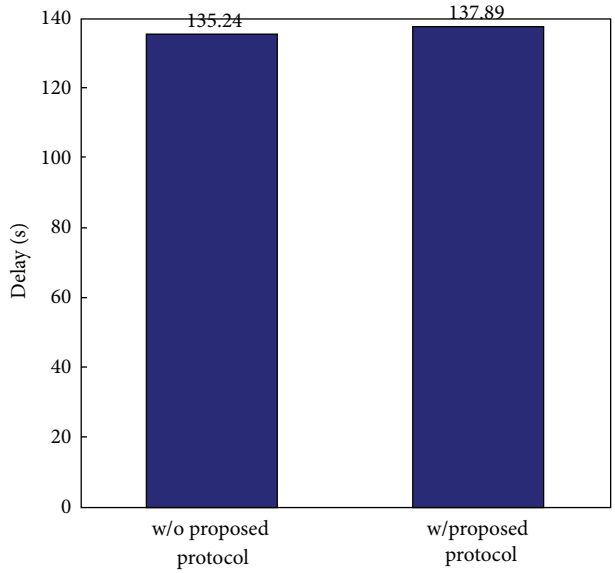


Figure 14: Delay performance of heterogeneous network.

detection methods to prevent attacks based on using artificial iris images.

Acknowledgment

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no. RGP-VPP-264.

References

[1] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[2] P. Kumar and H.-J. Lee, "Cryptanalysis on two user authentication protocols using smart card for wireless sensor networks," in *Proceedings of the Wireless Advanced (WiAd '11)*, pp. 241–245, London, UK, 2011.

[3] Z. Benenson, N. Gedicke, and O. Raivio, "Realizing robust user authentication in sensor networks," in *Real-World Wireless Sensor Networks (REALWSN)*, vol. 14, 2005.

[4] K. H. M. Wong, Z. Yuan, C. Jiannong, and W. Shengwei, "A dynamic user authentication scheme for wireless sensor networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pp. 244–251, Taichung, Taiwan, June 2006.

[5] H.-R. Tseng, R.-H. Jan, and W. Yang, "An improved dynamic user authentication scheme for wireless sensor networks," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 986–990, Washington, DC, USA, November 2007.

[6] L.-C. Ko, "A novel dynamic user authentication scheme for wireless sensor networks," in *Proceedings of the IEEE International Symposium on Wireless Communication Systems, ISWCS'08*, pp. 608–612, Reykjavik, Iceland, October 2008.

[7] B. Vaidya, J. J. Rodrigues, and J. H. Park, "User authentication schemes with pseudonymity for ubiquitous sensor network in NGN," *International Journal of Communication Systems*, vol. 23, no. 9-10, pp. 1201–1222, 2010.

[8] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.

[9] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu, "An enhanced two-factor user authentication scheme in wireless sensor networks," *Ad-Hoc and Sensor Wireless Networks*, vol. 10, no. 4, pp. 361–371, 2010.

[10] D. H. Nyang and M. K. Lee, "Improvement of Das's two-factor authentication protocol in wireless sensor networks," http://eprint.iacr.org/2009/631.pdf.

[11] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of "two-factor user authentication in wireless sensor networks"," *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.

[12] K. S. Arikumar and K. Thirumoorthy, "Improved user authentication in wireless sensor networks," in *Proceedings of the International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT '11)*, pp. 1010–1015, Tamil Nadu, India, March 2011.

[13] H.-L. Yeh, T.-H. Chen, P.-C. Liu, T.-H. Kim, and H.-W. Wei, "A secured authentication protocol for wireless sensor networks using Elliptic Curves Cryptography," *Sensors*, vol. 11, no. 5, pp. 4767–4779, 2011.

[14] J. Yuan, C. Jiang, and Z. Jiang, "A biometric-based user authentication for wireless sensor networks," *Wuhan University Journal of Natural Sciences*, vol. 15, no. 3, pp. 272–276, 2010.

[15] E.-J. Yoon and K.-Y. Yoo, "A new biometric-based user authentication scheme without using password for wireless sensor

networks," in *Proceedings of the 20th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 279–284, Paris, France, 2011.

[16] S. Mohammadi and S. Abedi, "ECC-based biometric signature: a new approach in electronic banking security," in *Proceedings of the International Symposium on Electronic Commerce and Security (ISECS '08)*, pp. 763–766, Guangzhou City, China, August 2008.

[17] A. Al-Hussain and I. Al-Rassan, "A biometric-based authentication system for web services mobile user," in *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia (MoMM '10)*, pp. 447–452, New York, NY, USA, November 2010.

[18] E. D. Leeuw, *Policies and Research in Identity Management: First IFIP WG11. 6 Working Conference on Policies and Research in Identity Management (IDMAN '07), RSM Erasmus University*, Springer, Rotterdam, The Netherlands, 2008.

[19] R. Watro, D. Kong, S.-F. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in *Proceedings of the 2004 ACM Workshop on Security of Ad Hoc and Sensor Networks, SASN'04*, pp. 59–64, New York, NY, USA, October 2004.

[20] M. R. Islam, M. S. Sayeed, and A. Samraj, "Biometric template protection using watermarking with hidden password encryption," in *Proceedings of the International Symposium on Information Technology (ITSim '08)*, vol. 1, pp. 1–8, 2008.

[21] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proceedings of the Second International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 162–175, New York, NY, USA, November 2004.

[22] L. Masek, "Recognition of human irispatterns for biometric identification," The University of Western Australia, http://people.csse.uwa.edu.au/pk/studentprojects/libor/LiborMasekThesis.pdf.

[23] Chinese Academy of Sciences, Institute of Automation, Database of 756 Greyscale Eye Images. Version 1. 0, 2003, http://www.sinobiometrics.com/.

[24] MATLAB simulator, http://www.mathworks.com/products/matlab/.

[25] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS '33)*, January 2000.

[26] G. Smaragdakis, I. Matta, and A. Bestavros, "SEP: a stable election protocol for clustered heterogeneous wireless sensor networks," Tech. Rep., Boston University Computer Science Department, 2004.

[27] X. H. Le, M. Khalid, R. Sankar, and S. Lee, "An efficient mutual authentication and access control scheme for wireless sensor networks in healthcare," *Journal of Networks*, vol. 6, no. 3, pp. 355–364, 2011.

[28] F. A. Aderohunmu, J. D. Deng, and M. Purvis, "Enhancing clustering in wireless sensor networks with energy heterogeneity," *International Journal of Business Data Communications and Networking*, vol. 7, no. 4, pp. 18–31, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 501594, 9 pages
http://dx.doi.org/10.1155/2013/501594

Research Article

Performance Evaluation of a Simple Cluster-Based Aggregation and Routing in Wireless Sensor Networks

Sung-Hwa Hong,[1] Jeong-Min Park,[2] and Joon-Min Gil[3]

[1] *Department of Maritime Information and Communications Engineering, Mokpo National Maritime University, 91 Haeyangdaehak-ro, Chukkyo-dong, Mokpo-si, Jeonnam 530-729, Republic of Korea*
[2] *Electronics and Telecommunication Research Institute, 218 Gajeong-ro, Yusung-gu, Daejeon-si, Chungnam 305-700, Republic of Korea*
[3] *School of Information Technology Engineering, Catholic University of Daegu, 13-13 Hayang-ro, Hayang-eup, Gyeongsan-si, Gyeongbuk 712-702, Republic of Korea*

Correspondence should be addressed to Joon-Min Gil; joonmin.gil@gmail.com

Received 28 January 2013; Accepted 22 April 2013

Academic Editor: Muhammad Khurram Khan

In future ubiquitous networks, sensor nodes should collect various environmental data and parameters. Because sensor nodes tend to have small and often irreplaceable batteries with limited power capacity, energy-efficient aggregation and routing are essential to achieve to a prolonged network lifetime. We propose a simple cluster-based data aggregation and routing algorithm (SCAR) that decreases the incurred overhead during the selection of cluster heads in wireless sensor networks. The performance results show that SCAR can prolong network lifetime via energy conservation and achieve energy-balancing when nodes are fixed or have limited mobility.

1. Introduction

1.1. Background. Recent advances in wireless communications and hardware techniques have made it possible to emerge the new networks called "wireless sensor networks (WSNs)" [1, 2]. The networks can be used mainly for monitoring the environments that human beings cannot access easily. However, because the networks have many limits as compared with traditional wireless networks, they cannot easily adopt the protocols used in traditional wireless networks. Particularly, due to a limited battery of each node, it is crucial for each node to consume its own energy efficiently and evenly [3].

In WSNs, a large number of sensor nodes and a sink node are deployed in spacious areas, and all of them are able to communicate with each other. Generally, every sensor node consists of processing units, diverse sensors, actuators, and an RF transceiver. Most of sensor nodes have capabilities to sense environments via sensors and to take measurements by controlling actuators. A WSN employs a sink node as a gateway device, which is used to link between its WSN and other networks such as Internet. Moreover, the sink node manages its network topologies and sensor nodes' tasks. The WSN can be used in many application areas for monitoring what is happening in certain areas and for managing services.

1.2. Motivation and Design Issues. Sensor networks complying with IEEE 802.15.4 will enable service users to access data or information whenever and wherever. In addition, routing transmission techniques using such a sensor have been sought in multilateral ways [4]. For WSNs, attribute-based addressing rather than global addressing is more appropriate [1, 2]. Since query characteristics generated in sensor works concern mostly attributes, attribute-based addressing is necessary. Thus, broadcasting or multicasting rather than point-to-point communications is more suitable. In addition, the data collected from sensor nodes are delivered to the user requesting it, by any of the various external media such as Internet or artificial satellite which are connected with nodes

called sink nodes. At this time, data aggregation is required to save the wasted energy attributed to redundant transmission of similar information among adjacent nodes in the process of delivering collected data.

Examining the characteristics of sensor networks, a cluster-based hierarchical routing algorithm seems to be ideal for WSNs. Such algorithm has several merits; it ensures more energy-efficient routing in the manner of forming local clusters and transmitting the information on events incurred in an adjacent area to the gateway nodes, followed by data aggregation by means of the gateway nodes. It also prevents inefficient query flooding through delivery to the gateway nodes for the requested query. To improve energy efficiency at the network layer, inefficient routing should be reduced and instead power control for sensor nodes can be added. It is possible to improve significantly energy efficiency by controlling sensor nodes to prevent them from being involving in transmission, for example, in sleep mode, with powering off their transceivers, although this technique is applicable to some kinds of applications only. Similarly, the technique of powering off transceivers except in the case of data transmission and receipt is used at the media access control (MAC) layer as well. At the MAC layer, however, any information on networks is insufficient and if necessary, it takes some time to power on and off transceivers, leading inevitably to delayed transmission. Since it is feasible to obtain information on the networks controlling transceiver power above the network layer, the power can be controlled with separation between some nodes where it is possible to transmit data to sink nodes and the other nodes where it is impossible to do so. As mentioned earlier, as it is possible to power off nodes not involved in transmission depending on the applications, its combination with an MAC layer protocol can enable it to achieve a more significant energy-saving effect.

Network-level communication protocols for WSNs can be classified into hierarchical and flat protocols. Hierarchical approaches, such as LEACH [5], TEEN [6], AP-TEEN [7], and MTE [8], use clustered structures to aggregate and route packets. Cluster heads collect data from their cluster members, aggregate it to reduce the amount of data to be transmitted, and transmit it to the sink node, a gateway to the user. Cluster members, which make up a majority of the nodes, can conserve energy, because they communicate only with their corresponding cluster heads and their distance from the head is generally short. Long-range direct transmission to the sink node, which requires high amounts of power, is only performed by a small number of cluster heads. Although these methods may significantly improve energy efficiency and network lifetime, they have their own limitations and drawbacks in practice. For example, they assume that all nodes in the sensor field, including the sink node, can communicate directly with one another. Additionally, they assume that each node can control the power. Without the first assumption, some nodes distant from the sink node may not be able to serve as a cluster head because they cannot transmit to the sink node. Moreover, because cluster heads are selected randomly, they may all end up being on one side of the network, although the probability of this happening is very small. In this case, the nodes on the opposite side have to transmit across the entire sensor field. Thus, the physical dimensions of the sensor field are limited by the transmission range of nodes. The second assumption is the basis for the argument that short-range transmission consumes less power than long-range transmission does. Without power control, the transmission of the same amount of data consumes the same amount of energy regardless of transmission distance. However, power control requires additional signaling between transmitter and receiver. Moreover, if a node generates small amounts of data infrequently, signaling overhead for power control (delay, power consumption) may be greater than that for the user data itself. More precise control requires more signaling because nodes are greatly influenced by small changes in environmental conditions such as temperature and humidity.

The algorithm proposed in this paper has the following properties. The designs of clustering and reclustering are simple, and clustering efficiency is considered a secondary problem. Cluster head selection is performed in a greedy manner via the local exchange of node energy states. Each cluster head determines when to abandon this role and become a cluster member, depending only on its own energy state. These local interactions and local decisions regarding clustering and reclustering reduce control overhead and increase scalability at the cost of reduced optimality. The clustered structure is not for routing purposes. Routing information is managed independently from cluster structure. A cluster member transmits packets only to its cluster heads, but a cluster head can transmit packets to any nodes that can route the packets to the sink node. This routing mechanism further simplifies the clustering process because gateway nodes for inter-cluster communication can be selected independently from other clusters.

The remainder of this paper is organized as follows. Section 2 presents a brief review of related work. In Section 3, we introduce our system model with network and energy models. In Section 4, we present the detailed operations of the proposed algorithm. Section 5 illustrates the impact of energy balancing on network connectivity using simulations. Finally, Section 6 concludes the paper.

2. Related Work

Supporting the mobility of sensor nodes is one of the most important factors to enable WSNs because wireless sensor nodes can be attached to the human body, vehicles, and other mobile objects. Hence, the network layer should be implemented with an efficient routing algorithm for such nodes [7, 9–13].

2.1. Hierarchical Routing Protocols. Direct communication is the simplest and the most intuitive way to send and collect sensor data. In a direct connection, each sensor node sends data to the base station directly. It is quite simple, but it may consume a large amount of energy for nodes farther away from the base station. Based on the first-order radio model [5], energy drains more rapidly as the distance

from the base station grows. To efficiently maintain the routing path between a sink node and sensor nodes, various routing algorithms have been proposed. The hierarchical routing algorithm is one of them. The typical representative algorithms are LEACH [5] and LEACH-C [14].

Heinzelman et al. [5] introduced clustering algorithm for sensor networks called low energy adaptive clustering hierarchy (LEACH). LEACH is a cluster-based protocol that includes distributed cluster formation. The authors allowed for a randomized rotation of the cluster head's role in the objective of reducing energy consumption (i.e., extending network lifetime) and to distribute the energy load evenly among the sensors in the network. Some variations of LEACH for energy saving have been reported in the literature. In [15], authors tried to improve the performance of LEACH in terms of energy-saving, by selecting cluster heads according to nodes' residual energy and distance with other cluster heads.

LEACH-centralized (LEACH-C) was proposed as an improvement of LEACH which uses a centralized clustering algorithm to create the clusters [14]. In LEACH-C, the base station collects the information on the position and energy level from all sensor nodes in the networks. Based on this information, the base station calculates the number of cluster heads and configures the network into clusters.

In contrast to LEACH, PEGASIS [16] organizes sensor nodes into a single chain. Messages are sent hop-by-hop along the chain starting with the node farthest from the base station. PEGASIS is often referred to as a chain-based protocol. The main advantage of this protocol is the low total energy dissipation, as nodes only need to communicate with their neighbors. It uses a linear programming model to generate the optimal cluster formation for extending the lifetime of a sensor network. However, PEGASIS assumes that every node has a global knowledge of the network, which is not feasible. Delays also increase as chains get longer. Spreading and collecting all sensors' information across a large network is often costly and impractical. Therefore, distributed clustering protocols are more desirable for large networks.

2.2. Other Routing Protocols. Various approaches have been recently reported in the literature. In [8], a new centralized algorithm for constructing the minimum total energy (MTE) chain was proposed. In each step of chain construction, the algorithm searches all remaining nodes and all possible insertion positions in the chain to select a node and a corresponding position in the chain that increases the total transmission cost of the chain. Then, the node is inserted into the chain at that position. MTE constructs chains with less total transmission energy cost than PEGASIS but is more computationally complex.

In [17], authors proposed the hybrid, energy-efficient, distributed (HEED) clustering protocol to prolong network lifetime and support scalable data aggregation. In this protocol, cluster heads are probabilistically selected based on their residual energy and the sensor nodes join the clusters according to their power level. HEED extends LEACH by incorporating communication range limits and cost information. In HEED, the clustering process is divided into a number of iterations. The cost can be either the node degree or the residual energy of a cluster head. Both HEED and LEACH can finish their executions within a constant number of iterations. To balance the energy consumption of all sensors, both protocols require reclustering after a period of time (called round), which causes extra energy consumption.

In [12], authors proposed hot spot-aware clustering approach based on two-tier hierarchy, in which cluster heads form the higher tier while member nodes form the low tiers. They address a hot spot problem that arises in the vicinity of the base stations. The unequal clustering mechanism that can evenly distribute energy consumption among nodes at different distances from the base station in WSNs was presented to solve the problem.

In [18], authors suggested a mixed algorithm with virtual gateway nodes, which includes both the advantages of existing hierarchical-structure algorithm and flat-structure algorithm in WSNs.

Our work is partly inspired by LEACH [5] and IGN [18]. In particular, IGN is divided into two classes: the primary class is to set a gateway-selection level and the secondary class is to offer a home automation with a routing technique, focusing on a sensor network to set a flooding level. However, the proposed scheme (SCAR) focuses on extending the lifetime of sensor networks with the integrated gateway node. Thus, it can provide an energy-efficient MAC protocol in network layer. Our scheme, together with the MAC protocol, can also be used for the enhancement of energy conservation and easily implementation.

3. System Model for Efficient Routing Protocol

3.1. WSN Clustering Environments. We use the same WSN environments as in [18]. Therefore, low-layer process with sensor nodes is also same. In most cases, it is reasonable to assume that sensor nodes have a fixed and relatively short transmission range. In this case, an energy-efficient multihop routing mechanism is essential, and cluster organization becomes more complex than in the single-hop condition. Efficient clustering algorithms for WSNs have to satisfy several requirements, such as [18] the following:

(1) clusters should cover entire sensor fields;

(2) average cluster size should be as large as possible to maximize data aggregation efficiency;

(3) clusters should be repeatedly reorganized to balance energy consumption among the nodes;

(4) clustering overhead should be small;

(5) clustering algorithms should be simple enough to be performed by low-performance processors with small available memory space.

A clustered structure organizes the sensor nodes into clusters, each governed by cluster head. The nodes in each cluster are involved in message exchanges with their respective cluster heads, and these heads send messages to a sink node, which is usually an access point connected to a wired network. Figure 1 represents a cluster architecture where

message can reach the sink node in a shorter hop. Clustering can extended to greater depths hierarchically.

A clustered structure is especially useful for sensor networks because of its inherent suitability for data fusion. The data gathered by all members of the cluster can be fused at the cluster head, and only the resulting information needs to be communicated to the sink node. Sensor networks should be self-organizing; hence, the cluster formation and selection of cluster head must be an autonomous, distributed process. This is achieved through network layer protocol such as the LEACH.

A clustered structure is very useful for conserving energy in a network [5]. The benefit comes from the data aggregation of cluster heads. Aggregation efficiency increases as more data packets are aggregated. This benefit, however, is limited in multihop networks because cluster size is limited by the radio transmission range of the nodes. On the other hand, clustering overhead increases as clustering becomes more complex. The complexity stems mainly from the fact that, in multihop networks, it is difficult to recluster in a synchronized, and when one cluster is reorganized (i.e., the role of a cluster head shifts from one node to another), the physical region that the cluster head covers is also changed. This may necessitate reorganization of other clusters to satisfy requirements 1 and 2. To better satisfy these requirements, however, more signaling and processing are required. However, requirements 4 and 5 prevent increasing the overhead and complexity [18].

Requirement 3 entails clustering overhead continually. However, if the nodes are mobile, this will greatly increase the overhead. Thus, the benefit of clustering can be reduced by the clustering overhead. In single-hop networks, node mobility does not affect any network operations as long as the node does not move out of the transmission range of any other node. Several network models for hierarchical protocols for WSNs, all single-hop networks, have been proposed [5–7]; those for flat routing protocols are all multihop networks [9, 10]. Clustering complexity in multihop networks can be one explanation for this research trend in sensor networks [11, 18–20].

Figure 1 shows our proposed WSN clustering system model. We assume that a WSN consists of N sensor nodes. The goal is to identify a set of cluster heads that cover the entire WSN. We denote a sensor node set by $S = \{s_1, s_2, \ldots, s_N\}$, where s_i represents the ith sensor ($i \in \{1, 2, \ldots, N\}$) and $|S| = N$. Each node s_i is then mapped to exactly one cluster C_j, where $j \in \{1, 2, \ldots, C\}$. C is the total number of clusters that covers all the nodes in the WSN ($C < N$). Each node in the WSN is mapped to exactly one cluster and must be able to directly communicate with its cluster head. Usually, communication within the cluster takes place over one-hop distance while traffic moves through the network over multiple hops to reach the base station.

In our model, every node can act as both a sensing source and a data-gathering source (cluster head), which motivates the need for efficient algorithms to select servers according to the system goals outlined later. A node only knows about the servers that are within its reachable range, which implies that achieving global goals cannot always be guaranteed but can be approximated through intelligent local decisions (given lower-layer flooding level in [18]). Finally, a node may fail if its energy resource is depleted, which motivates the need for rotating the server role among all nodes for load balancing.

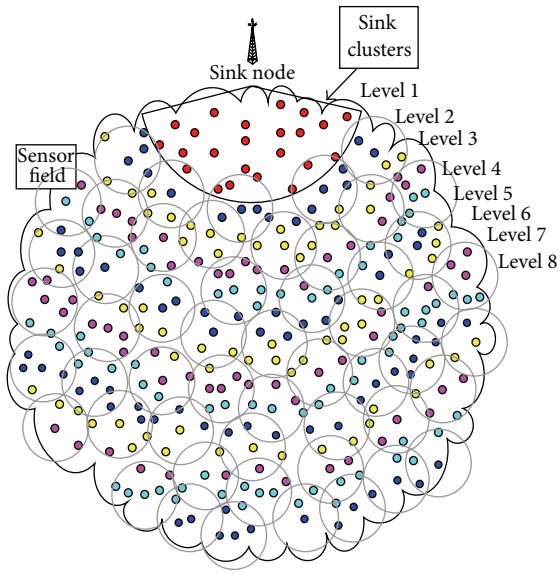


Figure 1: The proposed WSN clustering system model.

Energy conservation can be considered at the levels of individual nodes and the entire network. The energy efficiency of individual nodes is determined in various design stages including that of the circuit design, operating system, and medium access control (MAC), among others. The mechanism of energy conservation of each node has a significant impact on network lifetime. However, it is equally important to have an even distribution of energy consumption among nodes because the network may be unable to provide the required services if some nodes have used up their battery power, even if all other nodes have almost full batteries. Specifically, in multihop networks, one node failure can create one or more disconnects between other nodes and sink nodes, even if other nodes still have enough energy to operate. To prevent such situations, a network-level protocol is required to balance energy consumption among the nodes. In addition to energy consumption regulation, a network protocol should be able to achieve network-level energy conservation by reducing packet transmission via in-network data processing including data aggregation [1, 2, 18].

To improve energy efficiency at the network layer, inefficient routing should be reduced, and instead, a power control for sensor nodes can be added. It is possible to significantly improve energy efficiency by controlling sensor nodes, for example, by preventing them from being involved in transmission (e.g., entering sleep mode) and/or powering off their transceivers (for some applications). This latter technique can also be used at the MAC layer. However, it takes time to turn transceivers on and off, inevitably leading to delays in transmission. Because it is feasible to obtain information on networks controlling transceiver power above the network layer, the power can be controlled separately for

nodes that can transmit data to sink nodes and those that cannot.

3.2. Network Model. Based on the network model presented in [5, 18, 21], we assume our network model to be as follows.

(1) Many sensor nodes are dispersed randomly on a region of interest.

(2) Sink nodes are located at some convenient places in or near the sensor field. Users can obtain information from the sensor field and control it via sink nodes by direct or remote access. Thus, the sink nodes should have user interfaces or capabilities for communicating with remote users with high-powered radio or wired connections. There are very few sink nodes compared to the number of sensor nodes. Thus, they can be given special capabilities, such longer-lasting batteries or external power supplies.

(3) The sensor nodes have limited processing and communication capabilities. Thus, it is difficult to adopt very complex and/or highly energy-consuming algorithms

(4) All sensor nodes have the same constant transmission range. All nodes have similar capabilities (processing/communication) and equal significance. This motivates the need for extending the lifetime of every sensor

(5) Users request data from the sensor network by disseminating query packets through the sink nodes. The data sensed from each node are gathered by sink nodes through cluster heads so that users can access the data through the sink nodes.

(6) Clustering is completely distributed. Each node independently makes its decisions based only on local information.

(7) Clustering terminates within a fixed number of iterations (regardless of network diameter).

(8) Clustering should be efficient in terms of processing complexity and message exchange.

(9) Cluster heads are well distributed over the sensor field and have relatively high average residual energy compared to regular nodes.

Based on these characteristics, attribute-based addressing rather than global addressing is more appropriate and thus broadcasting or multicasting is more suitable than point-to-point communication. Cluster-based hierarchical routing is also suitable for the energy-saving of sensor nodes.

3.3. Energy Model. The parameters used in most energy models are similar to those used in LEACH [5]. Using the radio model presented in [5], the radio of power, $E_{T_x}(k, d)$, consumed by a transmitting node to send a k-bit message over distance d is

$$\begin{aligned} E_{T_x}(k, d) &= E_{T_x\text{-elec}}(k) + E_{T_x\text{-amp}}(k, d) \\ &= E_{\text{elec}} \times k + \varepsilon_{\text{amp}} \times k \times d^2. \end{aligned} \tag{1}$$

To receive this message, the radio of power ($E_{R_x}(k)$) expends

$$E_{R_x}(k) = E_{R_x\text{-elec}}(k) = E_{\text{elec}} \times k, \tag{2}$$

where $E_{\text{elec}} = E_{T_x\text{-elec}} = E_{R_x\text{-elec}}$.

In this paper, we use the typical values $E_{\text{elec}} = 50\,\text{nJ/bit}$ and $\varepsilon_{\text{amp}} = 100\,\text{nJ/bit/m}^2$. As described previously, cluster heads are responsible for aggregating their cluster members' data.

4. Simple Cluster-Based Aggregation and Routing

In this section, a routing algorithm for WSNs is proposed to improve the efficiency of energy consumption in sensor nodes. Each sensor node has the value called "Flooding Level" obtained through the initial flooding from a sink node instead of sending beacon messages in multihop sensor field. This value can be used for guaranteeing the sensor nodes to connect with a sink node and determining the roles of cluster-head and cluster-gateway node efficiently and simply during the clustering. The proposed algorithm, simple cluster based aggregation routing algorithm (SCAR), can provide the energy efficiencies in networks layer. This SCAR algorithm can be used together with existing energy-efficient MAC protocols to increase energy conservation. SCAR is divided into two parts, self-organization and routing algorithms.

4.1. Self-Organization for Cluster-Head Selection. The self-organization procedure consists of setting the routing information, an initial clustering, and then a reclustering procedure [22, 23].

4.1.1. Setting Routing Information. Routing information is flooded from the sink nodes. The procedure for each node to set the routing information for each sink node is similar to the distance vector algorithm. In the routing information packet, the number of hops to a specific sink node and the address of the transmitting node are included. When a node receives routing information from a neighbor node, it increases the number of hops by one and uses the number as its own number of hops to the sink node. Then, it retransmits this information with its own address. When different numbers of hops are received from different neighbor nodes, the smallest number is used. If a node receives a smaller number after it has retransmitted routing information, the smaller number should again be retransmitted to correct the propagated errors. Through this procedure, each node can determine its own number of hops to a specific sink and the address of the next hop node to the sink [18].

4.1.2. Initial Clustering. The initial clustering occurs during the initial distribution of routing information. In a routing information packet, information on the energy state of the transmitting node should be included. When a node has transmitted routing information, every neighbor of the node, except those who have previously transmitted the

information, will retransmit that information. In this way, each node can gather information about the energy states of every neighbor node and compare those energy states to its own. When a node has the local maximum amount of energy, it becomes a cluster head and broadcasts a cluster head advertisement (CHAD) message to its neighbors. Nodes that cannot become a cluster head wait for a CHAD message from other nodes for a predetermined period of time; a node that does not receive one repeats the exchange process of energy state information with other nonaffiliated nodes. This procedure is repeated until every node is affiliated with one cluster head. Any nonaffiliated node affiliates with the node whose CHAD message it receives first [24–27].

4.1.3. Gateway Selection. Each cluster head has one gateway node to be connected to a sink node. That is, cluster members are connected to a sink node through the cluster head at least within 3 hops. The gateway node may or may not be a cluster member of the cluster head that selects it. A cluster head sends a gateway selection (GWS) message to select its gateway node. Query dissemination from a sink node and data gathering to a sink node are performed through cluster heads and gateway nodes.

4.1.4. Reclustering. When a cluster head's energy decreases below a predefined threshold, it broadcasts a break up cluster (BUC) message to its neighbors and gives up its role as a cluster head. Cluster members that receive this message behave in one of the following two ways: (1) a node with another cluster head within one hop affiliates with that cluster head; (2) a node without a cluster head within one hop repeats the energy state information exchange process to elect a new cluster head, until no unaffiliated nodes remain. Figure 2 illustrates the reclustering procedure. Nodes "B" to "F" are formally affiliated with cluster head "A". When node "A" gives up the role of cluster head, the cluster members of "A" search for other cluster heads. Nodes "B" and "C" find a new cluster head in their respective one-hop range and affiliate with it. The remaining nodes start exchanging their energy states. Node "D" is elected as a new cluster head, and nodes "A", "C", and "E" become affiliated with node "D". In other cases, however, they can form two or more clusters depending on their energy states.

Newly organized clusters will generally have relatively small numbers because some of the former members can affiliate with other clusters, and moreover, the remaining nodes can be divided into smaller clusters. This degrades the data aggregation efficiency of the cluster, but greatly simplifies the reclustering process. To optimize cluster size for data aggregation efficiency, entire sensor fields may be reclustered. Moreover, cluster fragmentation is not permanent because small clusters will grow again when neighbor clusters are broken up.

4.2. Routing. Routing in sensor networks is different from traditional ad hoc networks and includes query dissemination and data gathering [11, 18].

4.2.1. Query Dissemination. Query dissemination is initiated from any sink node. When a sink node broadcasts a query packet, cluster heads within one hop of the sink node receive it and rebroadcast it to their member nodes. Whereas a non-gateway node just receives the query packet, a gateway node retransmits it to the cluster head that has selected it as a gateway. By repeating this procedure, all of the nodes in the sensor field can receive the query packet [18].

4.2.2. Data Gathering. Data are gathered by simply reversing the path of query dissemination. A cluster member node transmits its sensed data to its cluster head, which aggregates all received data with its own sensed data and retransmits them to its gateway. The gateway node forwards the data to its cluster head as an ordinary cluster member [18].

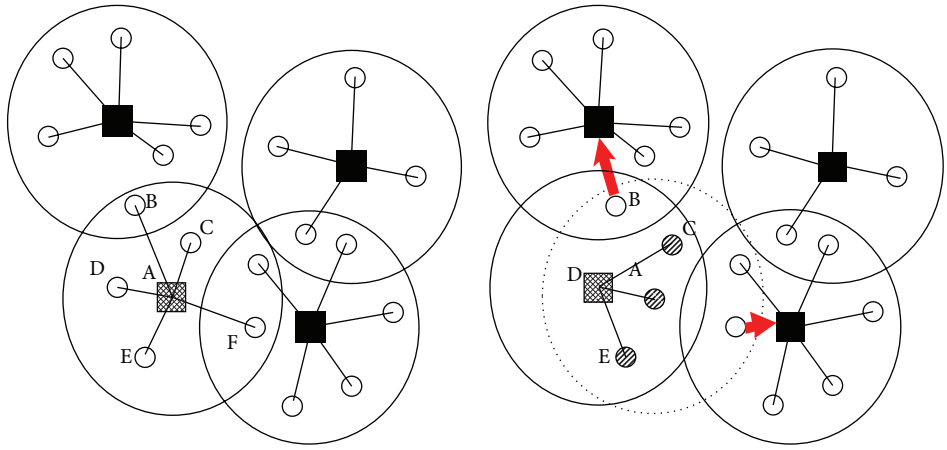
Once the power supply starts for data transmission, a control message that checks whether communications among neighboring sensor nodes are enabled is delivered. For the data flow in sensor networks, a query called "interest" requesting data transmission from a sink node to sensor nodes within the domain of the gateway node is transmitted to the closest gateway node. Then, the gateway node registers sensor nodes within its own domain as member nodes and delivers a specified address to each one. To transmit the data detected, the path from each node to the gateway node should be secured. Checking this requires receiving the minimal flooding signals originating from the sink node. In other words, any nodes failing to receive a flooding signal during this process would be isolated from neighboring nodes.

To check the connectivity of sensor nodes, flooding is necessary. Obviously, any energy that is consumed sending signals that are never received by any nodes is wasted. However, using this flooding process, sensor nodes can check that they are securely connected to the sink node, ensuring that their operations never waste energy and that they acquire the same network level as the sink node. Nodes that receive signals directly from the sink node set their own level to 1 and deliver the flooding message to neighboring nodes. Of the neighboring nodes that receive it, those that are not already set to a level of 1 set their own level to 2 and then again send the flooding message to neighboring nodes. This method makes it possible to check for the presence of neighboring nodes and at the same time secure connections with the sink node. Additionally, this technique can be used for cluster control. This process is shown in Figure 3.

5. Performance Evaluation

We evaluated the performance of the SCAR protocol via simulations and compared it with that of the MTE protocol [8], an energy-efficient flat routing protocol for multihop sensor networks.

We first examined the impact of the energy-balancing procedure on network connectivity defined as the ratio of the number of nodes with single-hop or multihop connectivity to the total number of nodes. Figure 4 shows the variation in connectivity depending on node density when node positions have a uniform random distribution in a square region. Two



(a) Before reclustering (b) After reclustering

Figure 2: Reclustering procedure.

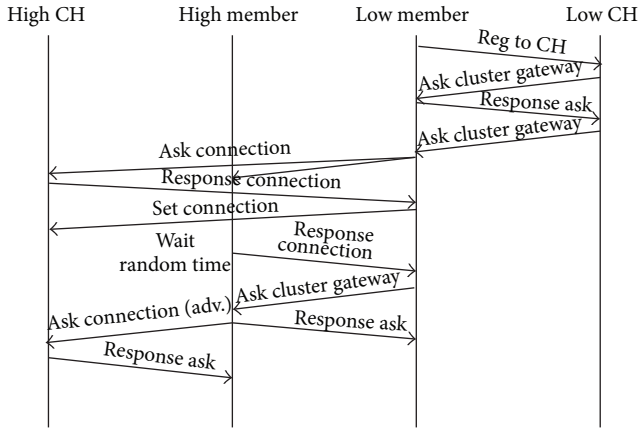


Figure 3: The link-setup processes of the cluster head.

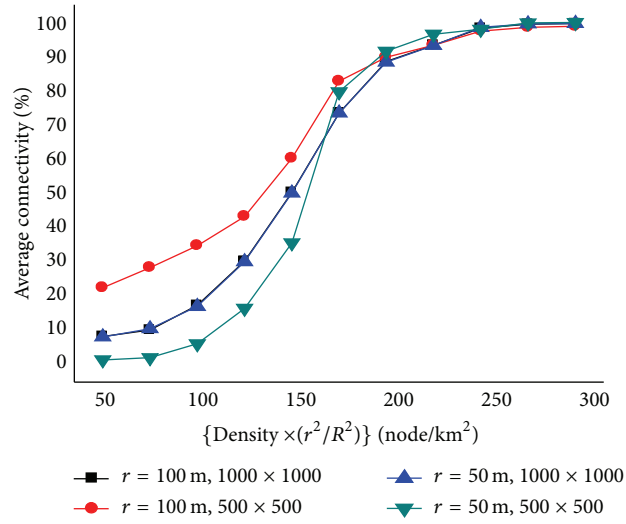


Figure 4: The effect of node density on network connectivity.

regions were designated: 500 × 500 m^2 and 1000 × 1000 m^2. We assumed that each sensor had a radio range of 50 m or 100 m. The simulation was repeated 100,000 times for each parameter vector (node density, radio range, region area).

Figure 4 can also be interpreted as an illustration of the effect of dead nodes on connectivity. For example, when nodes with a radio range of 50 m are deployed on a 0.25 km^2 (500 × 500 m^2) square region at a density of 200 nodes/km^2, the random death of 25% of the nodes (50/200) results in a disconnection (from the sink node) of roughly 55% of the nodes that are still alive. These nodes are useless, because the information sensed by these nodes cannot be delivered to the user. On the other hand, in a single-hop network, the death of one node eliminates just one node from the network. This explains why the balancing of energy consumption among nodes is especially important in multihop sensor networks.

The simulation scenario and the parameters are as follows. Initially, 100 nodes are dispersed randomly on a 500 × 500 m^2 region and a sink node is randomly selected from the nodes. The sink node initiates the routing information setup and initial clustering. Then, it transmits a query packet once every 100 ms and all nodes connected to the sink node respond to the query by transmitting a data packet.

Figure 5 illustrates the energy dynamics using SCAR and MTE. To illustrate the overhead of initial clustering, the initial energy is shown after initial clustering was finished. Due to the energy used for the first clustering, sensors in SCAR will start with less energy than will those in MTE. Indeed, in the simulations, SCAR initially had a lower energy than MTE for this reason. However, over time, SCAR gains energy efficiency. In MTE, the average remaining energy gradually decreases after 4 seconds. because the nodes start to die and become disconnected nodes, which no longer participate in data gathering and thus consume no more energy. Therefore, we can see from the results shown in Figure 5 that, in terms of data aggregation, the cluster structure of SCAR is superior to that of MTE.

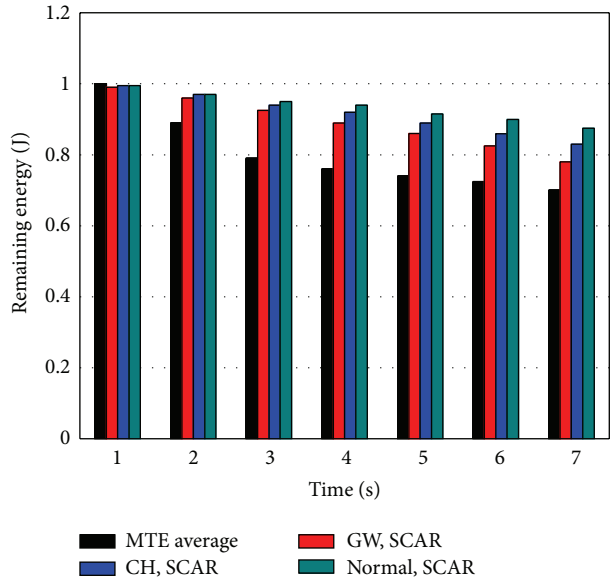


Figure 5: Average remaining energy over time.

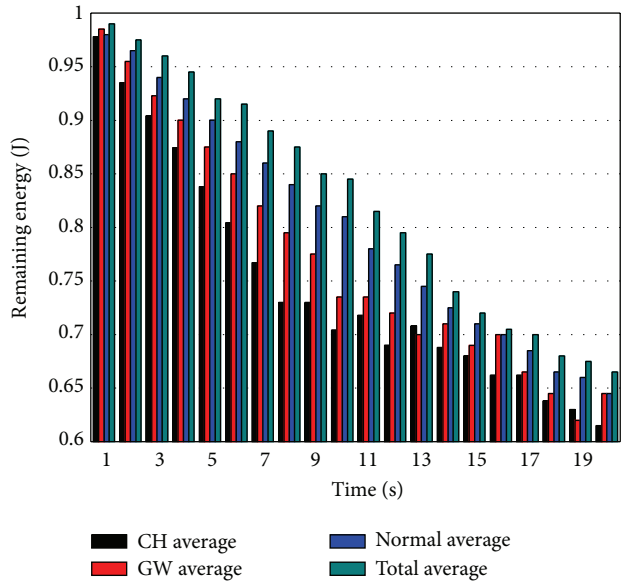


Figure 6: Average remaining energy for node role.

Next, we examined the energy efficiency of nodes in SCAR. Figure 6 shows the average remaining energy of nodes over time. The reduction in the energy of cluster heads begins to slow down after 7 seconds. due to energy-balancing. Some of the cluster heads give up their role and then become normal nodes. In this case, they consume more energy, and their energy falls below the threshold. On the other hand, the average remaining energy of normal nodes rapidly decreases from the initial stage, as some become cluster heads or gateway nodes. This local and nonperiodic reclustering is repeated continually. As a result, energy-balancing can be achieved in SCAR.

6. Conclusion

Our proposed algorithm, SCAR, is a simple, energy-efficient distributed clustering mechanism for multihop wireless sensor networks. It can decrease the incurred overhead during the selection of cluster heads in WSNs. Thus, the proposed algorithm enables efficient data aggregation, resulting in less data transmission and energy consumption. Performance results showed that our algorithm is much more efficient than MTE, an energy-efficient flat routing protocol, in terms of average energy consumption and network lifetime. Finally, our algorithm is appropriate for fixed and low-mobility networks. As a tradeoff for its simplicity, it can more or less lack an optimal cluster organization. Nevertheless, we showed that our algorithm operates effectively.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1A4A01015777).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor net-works: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] P. Rentala, R. Musunnuri, S. Gandham, and U. Saxena, "Survey on sensor networks," Tech. Rep. UTDCS-10-03, University of Texas, 2003.

[3] A. Kusdaryono and K. O. Lee, "A clustering protocol with mode selection for wireless sensor network," *Journal of Information Processing Systems*, vol. 7, no. 1, pp. 29–42, 2011.

[4] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'," *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.

[5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS-33 '00)*, pp. 1–10, January 2000.

[6] A. Manjeshwar and D. P. Agrawal, "TEEN: a routing protocols for enhanced efficiency on wireless sensor networks," in *Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, pp. 2009–2015, 2001.

[7] A. Manjeshwar and D. P. Agrawal, "APTEEN: a protocol for efficient routing and comprehensive information retrieval in wireless sensor networks," in *Proceedings of the 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, pp. 195–202, April 2002.

[8] K. Du, J. Wu, and D. Zhou, "Chain-based protocols for data broadcasting and gathering in the sensor networks," in *Proceedings of the International Parallel and Distributed Processing Symposium*, April 2003.

[9] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th Annual International*

Conference on Mobile Computing and Networking (MOBICOM '00), pp. 56–67, August 2000.

[10] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, vol. 8, no. 2-3, pp. 169–185, 2002.

[11] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '99)*, August 1999.

[12] M. K. Joshi, L. Osborne, B. Sun, and S. K. Makki, "Hot spot aware energy efficient clustering approach for Wireless Sensor Networks," in *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC '11)*, pp. 585–589, January 2011.

[13] P. D. Marco, P. Park, C. Fischione, and K. H. Johansson, "TREnD: a timely, reliable, energy-efficient and dynamic wsn protocol for control applications," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, pp. 1–6, May 2010.

[14] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[15] J. Chen, "Improvement of LEACH routing algorithm based on use of balanced energy in wireless sensor networks," *Lecture Notes in Computer Science*, vol. 6838, pp. 71–76, 2011.

[16] S. Lindsey and C. S. Raghavendra, "Pegasis: power-efficient gathering in sensor information systems," in *Proceedings of the IEEE Aerospace Conference*, March 2002.

[17] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[18] S.-H. Hong and B.-K. Kim, "An efficient data gathering routing protocol in sensor networks using the integrated gateway node," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 2, pp. 627–632, 2010.

[19] J. L. Wong and M. Potkonjak, "Search in sensor networks: challenges, techniques, and applications," in *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing*, pp. 13–17, May 2002.

[20] S. Park and D.-H. Cho, "Adaptive multicluster based cooperative communication in ad-hoc network," in *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC '12)*, pp. 14–17, January 2012.

[21] N. Xu, A. Huang, T.-W. Hou, and H.-H. Chen, "Coverage and connectivity guaranteed topology control algorithm for cluster-based wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 12, no. 1, pp. 23–32, 2012.

[22] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, 2000.

[23] Z. Haas and S. Tabrizi, "On some challenges and design choices in ad-hoc communications," in *Proceedings of the IEEE Military Communications Conference (MILCOM '98)*, October 1998.

[24] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 70–84, July 2001.

[25] A. Cerpa and D. Estrin, "ASCENT: adaptive self-configuring sensor networks topologies," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 272–285, 2004.

[26] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 481–494, 2002.

[27] Z. A. Aghbari, I. Kamel, and W. Elbaroni, "Energy-efficient distributed wireless sensor network scheme for cluster detection," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 28, no. 1, pp. 1–28, 2013.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 543950, 22 pages
http://dx.doi.org/10.1155/2013/543950

Research Article

A Performance and Usability Aware Secure Two-Factor User Authentication Scheme for Wireless Sensor Networks

Sang Guun Yoo, Hyukjun Lee, and Juho Kim

Department of Computer Science and Engineering, Sogang University, Seoul 121-712, Republic of Korea

Correspondence should be addressed to Juho Kim; jhkim@sogang.ac.kr

Received 23 January 2013; Accepted 22 March 2013

Academic Editor: Muhammad Khurram Khan

Recently, several user authentication schemes for wireless sensor networks based on two-factor concept using the smart card technology were proposed. However, they have serious limitations in terms of security and usability. First, even though they are enhancements of other works, they still have several security flaws, such as vulnerability against parallel session, privileged-insider, and gateway-node bypassing attacks and lack of mutual authentication between user station and gateway node. On the other hand, they also present a usability constraint, in a sense that they do not consider the use case when sensor nodes cannot communicate with gateway node. In this case, data collected by isolated sensor nodes could not be accessed until they recover such communication, which is in many times not recoverable rapidly or forever (e.g., military applications, natural disaster monitoring). Due to all these reasons, this paper proposes a robust user authentication scheme which fixes the security weaknesses of previous solutions and provides wider usability considering the use case when the sensor nodes cannot communicate with the gateway node. Once the solution is described, its security is ensured by formal proof and analysis against attacks. Additionally, performance and cost analysis are executed to determine its level of feasibility for real implementation.

1. Introduction

With the growth of wireless sensor network (WSN) application fields, the frequency of WSNs managing critical tasks and important information has also increased. However, most of real applications do not include any security mechanisms making them vulnerable to serious attacks. This fact makes evident the necessity of security solutions for WSNs such as [1]. Among different security mechanisms, a user authentication mechanism that allows only legitimate users to access the WSN's data is considered one of the most important security mechanisms because it contributes to keeping the confidentiality and integrity of network's data and because it is an essential primitive upon which other security mechanisms such as secure channel establishment and over-the-air (OTA) programming [2] are built on.

At this point, several research works [3–7] have been executed for providing user authentication schemes for WSNs. Those proposals consider the resource limitations of WSNs and provide lightweight schemes. However, they have serious limitations in terms of security and usability. First, previous works have several security flaws such as vulnerability against parallel session, privileged-insider, and gateway-node bypassing attacks and lack of mutual authentication between user station and the gateway node. On the other hand, they also present a usability constraint, in a sense that they do not consider the case when sensor nodes cannot communicate with the gateway node; in this case, data collected by sensor nodes could not be accessed until they recover this communication, which is in many times not recoverable rapidly or forever. (e.g., military applications in war, natural disaster monitoring). Due to all these reasons, this paper proposes an enhanced user authentication scheme that solves the identified security issues and constraints.

The rest of this paper is organized as follows. Section 2 analyzes the existing works to detail their weaknesses and limitations in terms of security and usability. Section 3 then specifies the design criteria of the proposed user authentication scheme. Later, Section 4 presents the details of the proposed solution which solves the vulnerabilities and limitations mentioned in Section 2. Next, Section 5 analyzes the proposed protocol in terms of achieved security

Table 1: Summary of cryptanalysis of previous works based on smart cards.

Security feature	Das' [3]	Nyang-Lee's [4]	Huang et al.'s [5]	Chen-Shih's [6]	Khan-Alghathbar's [7]
Secure password change/update	No	No	Yes	No	Yes
Protection against insider's attack	No	No	No	No	Yes
Protection against gateway bypassing attack	No	No	No	No	Partial
Protection against parallel session attack	No	Yes	No	No	No
Mutual authentication (gateway-sensor node)	No	Yes	No	No	Yes
Mutual authentication (user station-gateway)	No	Yes	No	No	No
Session key establishment	No	Yes	No	No	No

level, performance, and cost. Finally, Section 6 details the conclusions of this paper.

2. Analysis of Existing Works

Lately, user authentication for wireless sensor networks based on smart cards has been actively researched. The two-factor authentication approach which requires the verification of ownership of both a password and a smart card achieves effectively the purpose of authentic delivery of sensed data while minimizing the load of storing user data in the gateway node. This is because the tamper-proof smart card of the user delivers the function of secure storage of authentication data instead of the gateway node.

In 2009, Das [3] presented a research work where The author proposed an authentication scheme based on the two-factor user authentication concept using the smart card technology. Das' proposal was considered as an efficient two-factor user authentication scheme because it only required the usage of small number of hash function calculations [4]. However, even though Das' proposal was adopted by different research works, its limitations and security flaws were discovered in subsequent works. Nyang and Lee [4] identified that Das' protocol was vulnerable to offline password guessing and sensor node compromising attacks. Huang et al. [5] also identified some limitations of Das' scheme such as vulnerability from impersonate attack. Additionally, the authors of [6] pointed out the absence of mutual authentication feature in Das' protocol while Khan and Alghathbar [7] pointed out additional security flaws of Das' proposal, describing that it was vulnerable to privileged-insider and gateway-node bypassing attacks.

Once they discovered different vulnerabilities and security limitations of Das' proposal, the authors of [3–6] also proposed enhanced versions of Das' protocol to eliminate detected vulnerabilities. However, as shown in our previous work [8], those protocols still include serious vulnerabilities and security limitations of which an attacker can take advantage, exposing WSNs to serious risks. Table 1 summarizes the security analysis of [8] showing that the protocols in [3–7] are still vulnerable to different attacks and have several security limitations.

In addition, all of the aforementioned approaches only focus their analysis on the security and performance aspects, and neglect the usability aspect. From this point of view, we can say that the previous user authentication schemes

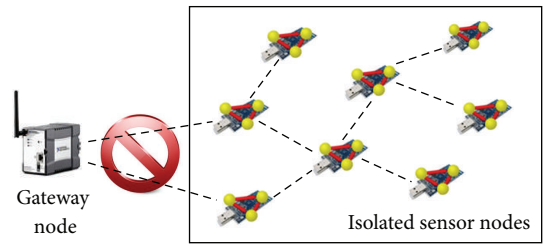


Figure 1: Sensor nodes isolated because of broken link between gateway and sensor nodes.

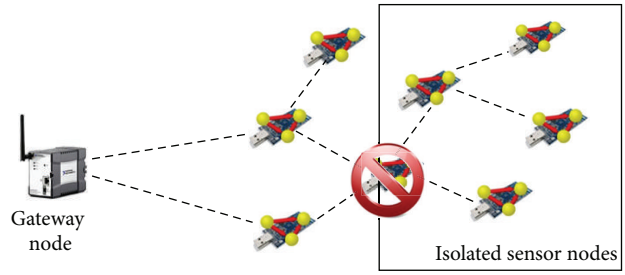


Figure 2: Sensor nodes isolated because of broken link between sensor nodes.

also present a serious usability constraint because they do not consider the case when important sensor nodes are isolated. Isolation of sensor nodes could occur because of network link failures between the gateway node and sensor nodes (see Figure 1) or because of disconnections between critical routing sensor nodes (see Figure 2). Sensor nodes isolation is considered problematic for several reasons. First, the isolated sensor nodes frequently store critical information which requires to be transmitted opportunely to the user for decision making. However, using the previous user authentication mechanisms, users could not authenticate to the isolated sensor nodes until sensor nodes recover the communication with the gateway node, which is in many times not recoverable rapidly or forever. Additionally, if the link between the gateway node and isolated sensor nodes is not reestablished rapidly, historical data of those nodes could be eliminated because of their limited storage memory capacity. In this regard, this situation demands for a new user authentication scheme with an offline user authentication mechanism which allows users to authenticate directly to

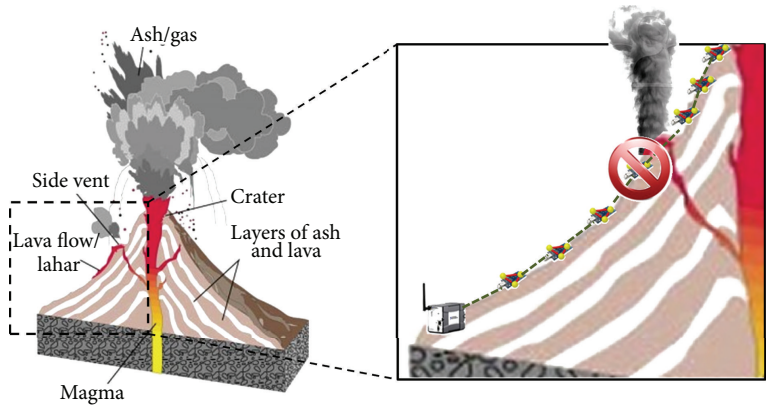
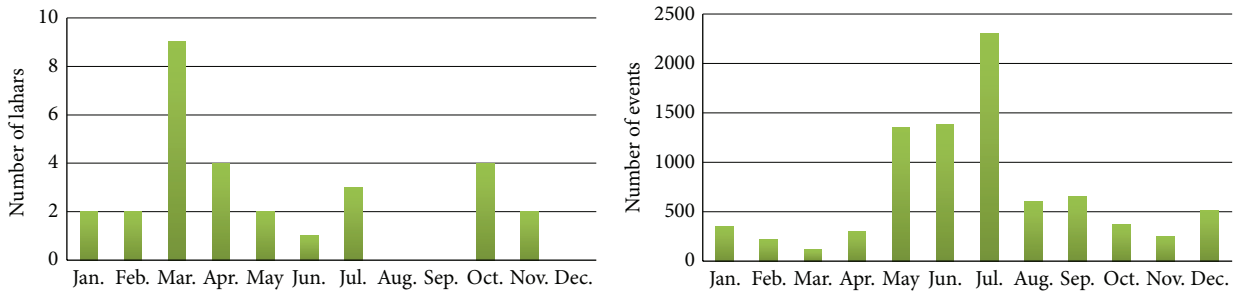


Figure 3: Sensor nodes isolated because of side vent explosion of a volcano.



(a) Number of lahars

(b) Number of long period seismicity

Figure 4: Statistical data of Tungurahua volcano in 2004.

isolated sensor nodes for getting the critical information in an opportune way.

Even though many people may think sensor node isolations are not common, they are common in particular applications. Here we describe some applications where sensor nodes could be isolated and why offline user authentication (mobile user station authenticating directly to the sensor node) is important. First, let us consider a volcano monitoring sensor network [9] gathering seismic and infrasonic signals. In such systems, there are several events that could provoke sensor nodes' isolation. One of the possibilities is when an explosion occurs in a side vent (see Figure 3); in such case, the sensor nodes located in the superior region of the side vent can lose the connection with the inferior region because the sensor nodes nearby the side vent are destroyed or buried. Another case is when critical routing sensor nodes are buried or damaged because of lahars or seismic activities. Data of Tungurahua volcano in Ecuador (see Figure 4) published by Instituto Geofísico de la Escuela Politécnica Nacional (http://www.igepn.edu.ec/) illustrates how frequent this situation could occur in an active volcano. The report indicates that 29 lahars and 8400 long-period seismic events were detected only in 2004. In those cases, it is frequent that the isolated nodes store important data that could help to forecast the future behavior of the volcano. Therefore, it is important to provide a mechanism that allows a rapid and opportune access to such information while maintaining the confidentiality and integrity features.

Another type of applications where critical sensor nodes could be isolated is the military one. Consider that a battlefield application has lost some intermediate nodes because they have been destroyed by the enemy, but the nodes deployed in the enemy's territory have accumulated important data. In this case, an automaton user station could be sent to the enemy's territory to gather such information. An easy way to authenticate to the sensor nodes is that the automaton carries the secret keys. However, carrying the secret keys in hostile environment opens wide possibilities of leakage of the secret values which could compromise the security of the whole network. Therefore, the user authentication scheme must provide a way to maintain the security of the sensor network even if the automaton is taken by the adversary.

In conclusion, as described previously, previous works present several limitations in terms of security and usability, and this situation creates the need for designing an enhanced user authentication scheme which overcomes such constraints.

3. User Authentication Scheme Design Criteria

We believe that several limitations of previous works were produced because of the absence of a concrete and clear requirement elicitation process. To avoid falling into the same mistake, this paper has decided to describe the design criteria of the proposed user authentication scheme.

3.1. General Considerations and Assumptions. The proposed user authentication scheme must be able to be implemented in a scenario with the following considerations and assumptions.

(i) The network is composed of traditional elements, that is, sensor nodes, gateway node(s), and user station with time synchronization.

(ii) The network has at least one gateway node which has a stable link with the user station.

(iii) The network implements a routing protocol which provides the path between sensor nodes and gateway node(s).

(iv) The hierarchies of sensor nodes and the topology of the network are taken care of by the routing protocol.

(v) Each sensor node can store a set of predefined data.

3.2. Security Requirements. The most important aspect of the proposed user authentication scheme is its security. Therefore, it is vital to precisely define the security requirements that the proposed scheme must satisfy. This section defines the threat model and then describes the list of security requirements considered in this paper.

3.2.1. Threat Model. In the analysis of the proposed protocol, the widely used Doley and Yao [10] threat model will be used, which assumes that two communicating parties communicate over an insecure channel. This means that the attacker is able to eavesdrop and manipulate the messages sent over the air. In other words, the attacker can read, modify, and delay the messages sent by the different entities that participate in WSNs. Additionally, it is assumed that the attacker may know the algorithm of the security mechanisms that are deployed in the network. The main aim of the attacker is the falsification of an authentication. It is assumed that the attacker aims for a forgery in the proposed message exchange scenario where a user authenticates to the sensor network. The attack is considered successful if the sensor network (whether sensor node or gateway node) accepts a fake message that was not sent by an authentic user or if the user accepts messages coming from a fake sensor node or gateway node. All kinds of nonauthentic messages such as random new messages, replayed messages, and modified messages sent by the attacker are considered as fake messages.

This paper only takes care of the user authentication problem; other security issues of other layers of the protocol are taken care of by security solutions of other levels. In addition, it is assumed that gateway node(s) is managed by a trusted infrastructure. Therefore, the gateway node(s) is considered as secure and its security is not considered as part of this work. Additionally, not every aspect of physical attacks is considered in this paper; however, the unauthorized extraction of the secret values of a sensor node or smart card using techniques such as shown in [11–13] must not have effect on the security of the rest of nodes and users.

3.2.2. Basic Requirements

Data Confidentiality. Data confidentiality is the most important issue in network security. The proposed security solution must provide concealment of private information making it infeasible for an unauthorized user to understand the confidential data.

Data Integrity. With the implementation of confidentiality, an adversary may be unable to read the information. However, this does not mean the data is safe. The adversary can change the data to produce disorder in the sensor network. Therefore, it is important that different entities of the network can detect modification of messages transmitted over the network.

Data Freshness. Even if confidentiality and data integrity are assured, it is necessary to ensure the freshness of each message. Data freshness suggests that the data is recent and it ensures that no old messages have been replayed.

3.2.3. Attack List. There are several attacks that have been considered common in user authentication for WSNs such as privileged-insider, stolen-verifier, replay, parallel session, guessing, brute force, impersonation, and gateway-node bypassing attacks [3–7]. The proposed solution must also demonstrate its security against such attacks.

3.2.4. Other Security Requirements

Mutual Authentication. Some of previous works [4, 7] provide mutual authentication between the gateway node and sensor nodes but do not provide mutual authentication between user and gateway node. This situation can compromise the security of the whole network because newer sensor network implementations offer remote administration/query features in their gateway nodes [14, 15], allowing users to access to network's data from a remote terminal. In this kind of environment, it is really important to authenticate the validity of the gateway node from the user's side to avoid adversaries collecting valuable data using fake gateway nodes.

Secure Registration, Authentication, and Password Change Processes. It is important to remember that the proposed user authentication mechanism must offer user registration, authentication, and password change processes and they must be executed in a secure manner.

Session Key Establishment. After authentication, the scheme must provide a simple session key establishment algorithm to provide a secure channel between entities after authentication.

3.3. Performance Requirements. WSNs own special characteristics which must also be considered in the designing of the security mechanisms. The most important resource constraints of WSN are the resource related ones [16], namely, limited storage space, limited computation power, and low energy capacity. Since the use of security algorithms reduces the lifetime of nodes, it is critical that the user authentication algorithm uses low quantity of energy. The aspects to be considered as performance requirements are as follows.

Type of Cryptographic Algorithm. Traditionally, there are two different types of cryptography algorithms, namely, public-key cryptography and symmetric-key cryptography. The first one has the characteristic of using two separate keys: one to encrypt the plaintext and another to decrypt the ciphertext. The most representative algorithms in public-key cryptography are Rivest-Shamir-Adleman ((RSA) the creators of the algorithm) and elliptic curve cryptography (ECC). On the other hand, the symmetric-key cryptography uses a common key for encryption and decryption which is shared among the communicating parties. One of the most representative algorithms of symmetric-key cryptography is advanced encryption standard (AES). Additionally, there is another type of encryption/decryption mechanism based on hash functions and exclusive-or operations which is considered as part of symmetric-key cryptography because it uses a common key between the communicating entities. One of the most representative hash functions used in such mechanisms is the secure hash algorithm (SHA). The importance of selecting a correct cryptographic algorithm lies in its complexity. A different Different cryptographic algorithms mean different complexities which is reflected in the required computation power and energy usage. This issue is very important because the common sensor nodes have very limited computation power and energy capacity. Several research works [17–20] have applied the asymmetric-key algorithms in WSNs. The results of such works reveal that despite the use of energy efficient techniques, such as ECC or dedicated cryptography coprocessors, asymmetric-key algorithms consume more energy than symmetric-key algorithms. For this reason, many researchers believe that the processing time and power consumption make it undesirable for public key algorithm techniques to be employed in sensor networks. Based on this criterion, the public cryptography was discarded for the proposed solution.

On the other hand, between the traditional symmetric key cryptography and encryption using hash functions, this paper has considered to use the last one because of its benefits in terms of energy usage as shown in previous works such as [21, 22]. In case of [21], the authors explain how AES-128 consumes more than double of SHA-1 function showing that the last one consumes around 154 μJ while AES-128 consumes 339 μJ on CrossBow nodes. The work described in [22] shows how for payloads of 17 bytes or above SHA-1 requires considerably less iterations than AES and therefore a shorter running time and less energy. In another earlier publication [16], the authors estimated that the energy per bit consumed by MIPS R4400 and MC68328 "DragonBall" processors for performing AES encryption/decryption operations is 9 nJ/bit and 101 nJ/bit, respectively, while for SHA-1 hashing function, the same processors consume 7.2 nJ/bit and 41 nJ/bit, respectively.

Number of Cryptographic Operations. The number of cryptographic operations used in sensor nodes must be minimal to extend their lifetime. The number of cryptographic operations executed by the user station and gateway node is not considered as important because they own superior resource capacities (i.e., computation power, storage, and energy capacity).

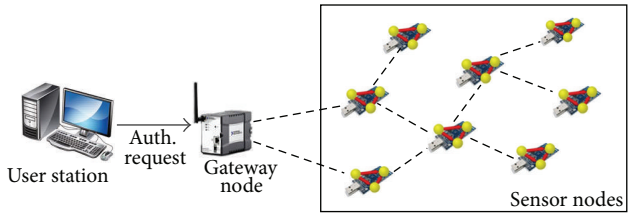


Figure 5: Online user authentication.

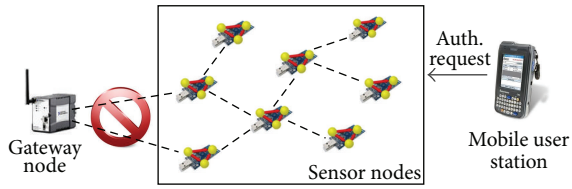


Figure 6: Offline or gateway-less user authentication.

Number of Messages. One of main the operations that consume more energy is the transmission of messages. Therefore, the number of messages sent by sensor nodes must also be minimal to reduce their energy spent in wireless communication.

Energy Consumption Analysis. Energy consumption analysis using realistic data sizes will help to understand the effects of the proposed user authentication scheme over WSN.

3.4. Usability Requirements. The proposed scheme must also consider the different use cases of the authentication process according to the state of the network to offer extensive usability. This paper considers two specific use cases.

3.4.1. Online User Authentication. Online user authentication refers to the user authentication process executed when the sensor node has network connection with the gateway node (see Figure 5). This is considered the most traditional because sensor nodes are commonly monitored from the fixed network infrastructure installed in a safe place which includes the gateway node.

3.4.2. Offline or Gateway-Less User Authentication. This case is when the user authenticates directly to the sensor node because the network connection between the sensor node and gateway node cannot be established. In this case, the user must approach to the sensor node to authenticate with it but without the authorization of the gateway node (see Figure 6).

Important. Although this use case is very important for different types of WSNs such as military operation and natural disaster monitoring applications (as explained in Section 2), none of the previous works have considered this use case.

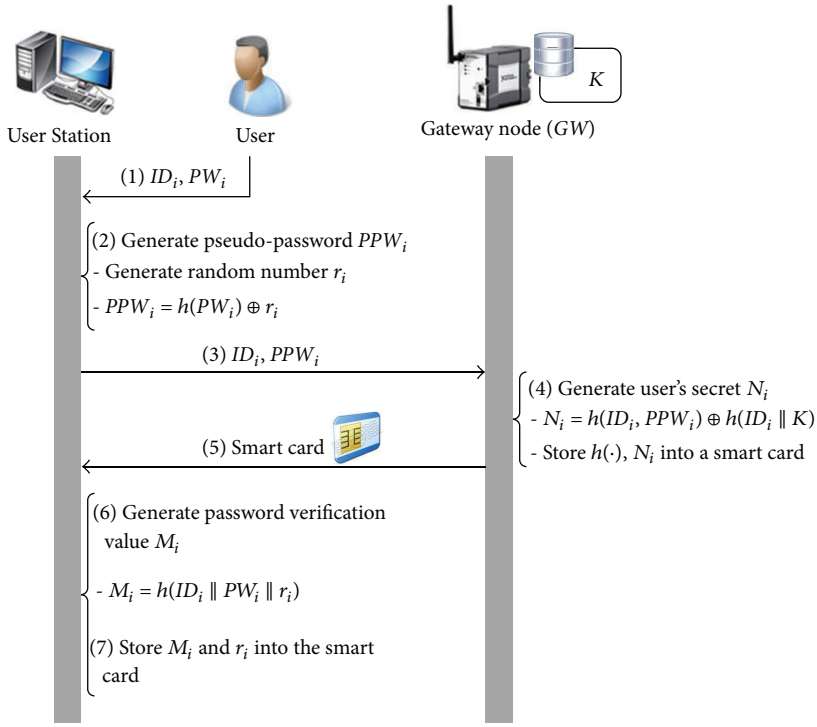


Figure 7: Proposed user registration protocol.

4. Proposed User Authentication Scheme

The proposed solution is composed of three protocols: user registration, user authentication, and password change protocols which are executed among three independent entities, that is, users, gateway node, and sensor nodes.

4.1. User Registration Protocol. The user registration protocol is executed when new user needs to be authorized to access the sensor network. The steps executed in this protocol are as follows (see Figure 7 and Table 2). A user U_i chooses his/her identity ID_i and password PW_i and inputs them to the terminal. The terminal then generates a random number r_i and computes $PPW_i = h(PW_i) \oplus r_i$, where $h(\cdot)$ is a hash function and $\oplus$ is an XOR operator. Once PPW_i has been calculated, ID_i and PPW_i are sent to the gateway node GW. GW then computes $N_i = h(ID_i \| PPW_i) \oplus h(ID_i \| K)$, where K is a symmetric key only known by GW and "$\|$" is a concatenation operator. Once N_i has been calculated, GW personalizes a smart card with the parameters $h(\cdot)$ and N_i. Then, GW delivers the smart card to U_i in a secure manner. Finally, U_i calculates $M_i = h(ID_i \| PW_i \| r_i)$ and stores M_i and r_i into the smart card.

On the other hand, a unique secret key $K_n = h(S_n \| K)$ is stored in each sensor node responsible for exchanging data with U_i, where S_n is the unique identification of the sensor node.

The proposed user registration protocol includes several enhancements compared to the previous solutions. First, each smart card and sensor node includes unique secret values: $N_i = h(ID_i \| PPW_i) \oplus (ID_i \| K)$ and $K_n = h(S_n \| K)$, respectively, which limits the effect of the physical attack to the attacked unit, maintaining the security of the rest of the network. Additionally, the usage of PPW_i (instead of PW_i) provides protection against the privileged-insider

Table 2: Notations of the proposed scheme.

Notation	Description
ID_i	Identification of user i
PW_i	Password of user i
$NewPW_i$	New password of user i
r_i	Random number of user i
K	Secret key of GW
K_n	Secret key of S_n
$T_1, T_2, T_3, T_4, T_{permit}$	Timestamps
$Permit_{in}$	Permit of user i to access S_n
$K_{U_i\text{-}GW}$	Session key between U_i and GW
$K_{S_n\text{-}GW}$	Session key between S_n and GW
$K_{U_i\text{-}S_n}$	Session key between U_i and S_n
$E_x(y)$	Symmetric encryption of string y using the key x
$D_x(y)$	Symmetric decryption of string y using the key x
$h(\cdot)$	One-way hash function
$\|$	String concatenation
$\oplus$	XOR operation
?=	Comparison operation

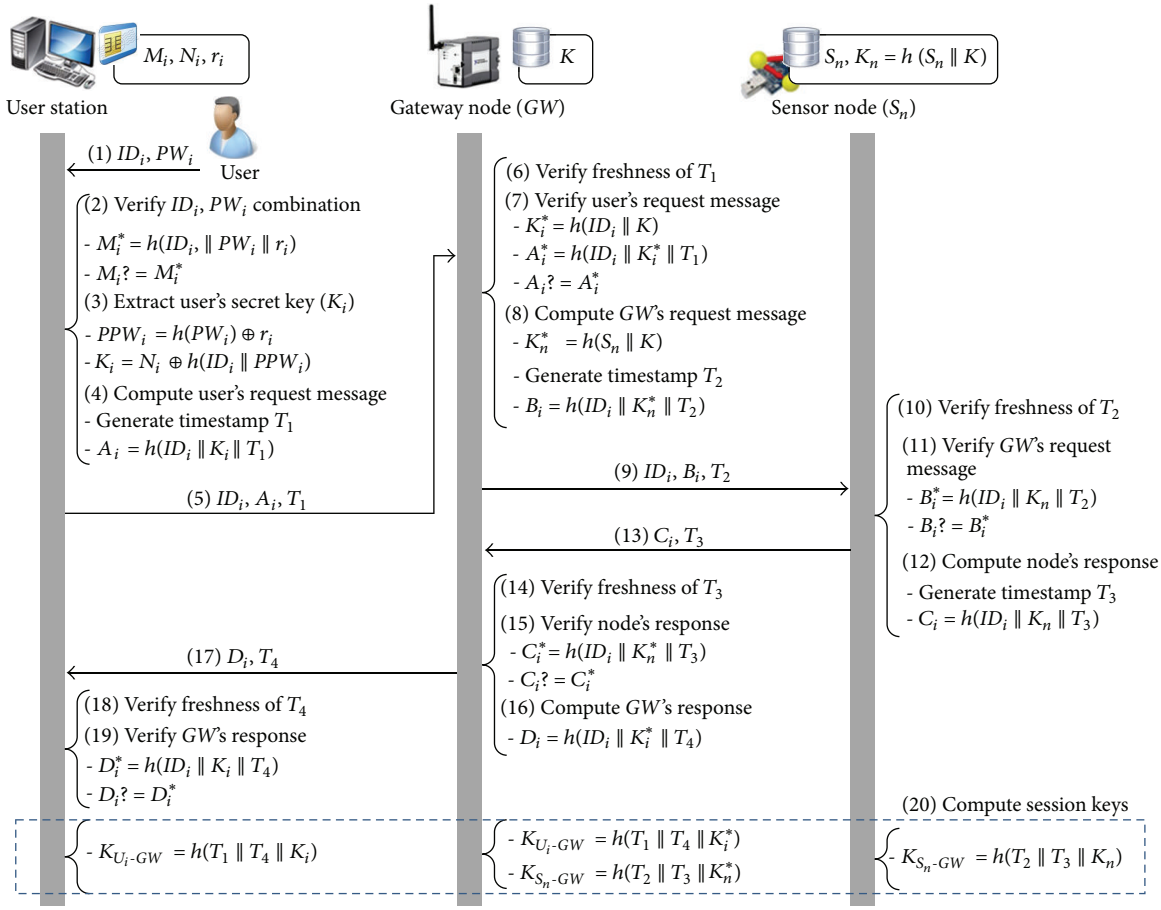


Figure 8: Proposed online user authentication protocol.

attack. Finally, the usage of $h(ID_i \| K)$ inside N_i delivers protection against parallel session attacks.

4.2. User Authentication Protocol. This protocol is performed when U_i needs to access the data gathered by a sensor node. The process of user authentication is differentiated depending on the use case as described in Section 3.

4.2.1. Online User Authentication Protocol. This subsection describes the authentication protocol when the sensor node is connected to the gateway node (see Figure 5). In this case, the gateway node works as a verifier to validate the authenticity of the user and sensor node. The steps executed in this protocol are as follows (see Figure 8 and Table 2).

Authentication Phase. U_i inserts the smart card and inputs his/her ID_i and PW_i. The smart card then computes $M_i^* = h(ID_i \| PW_i \| r_i)$ and compares M_i^* with M_i to authenticate U_i. If those values do not match, the authentication request is rejected. Otherwise, the smart card computes $PPW_i = h(PW_i) \oplus r_i$, $K_i = N_i \oplus h(ID_i \| PPW_i)$, and $A_i = h(ID_i \| K_i \| T_1)$ and transmits $\{ID_i, A_i, T_1\}$ to GW, where T_1 is the current timestamp of U_i's system. Upon receiving the login request at time T_1^*, GW validates T_1. If $(T_1^* - T_1) > \Delta T$, GW aborts the authentication process, where ΔT denotes the maximum allowed communication delay. Otherwise, GW computes $K_i^* = h(ID_i \| K)$ and $A_i^* = h(ID_i \| K_i^* \| T_1)$. Once A_i^* is computed, GW checks whether it is equal to A_i. If A_i^* is different to A_i, GW finishes the authentication process; otherwise, GW computes $K_n^* = h(S_n \| K)$ and $B_i = h(ID_i \| K_n^* \| T_2)$ and transmits the message $\{ID_i, B_i, T_2\}$ to some nearest sensor node S_n to respond to the query with the data that U_i is looking for, where T_2 is the timestamp of GW's system when sending the message. S_n first validates T_2 using similar method of T_1 verification then computes $B_i^* = h(ID_i \| K_n \| T_2)$ and checks whether B_i^* is equal to B_i. Only if those values match, S_n responds to GW's message by sending the message $\{C_i, T_3\}$, where $C_i = h(ID_i \| K_n \| T_3)$. Once GW is received the message, validates T_3 using similar method of T_1 and T_2 verification and then checks the validity of C_i by comparing $C_i^* = h(ID_i \| K_n^* \| T_3)$ with the received C_i. Once the sensor node is authenticated, GW computes $D_i = h(ID_i \| K_i^* \| T_4)$, where T_4 is the current timestamp of GW and sends the message $\{D_i, T_4\}$ to U_i. Finally, U_i validates T_4 and computes $D_i^* = h(ID_i \| K_i \| T_4)$ and checks whether it is equal to D_i.

Session Key Establishment Phase. A session key between U_i and $GW K_{U_i\text{-}GW} = h(T_1 \| T_4 \| K_i)$ and a session key between GW and $S_n K_{S_n\text{-}GW} = h(T_2 \| T_3 \| K_n)$ could be used

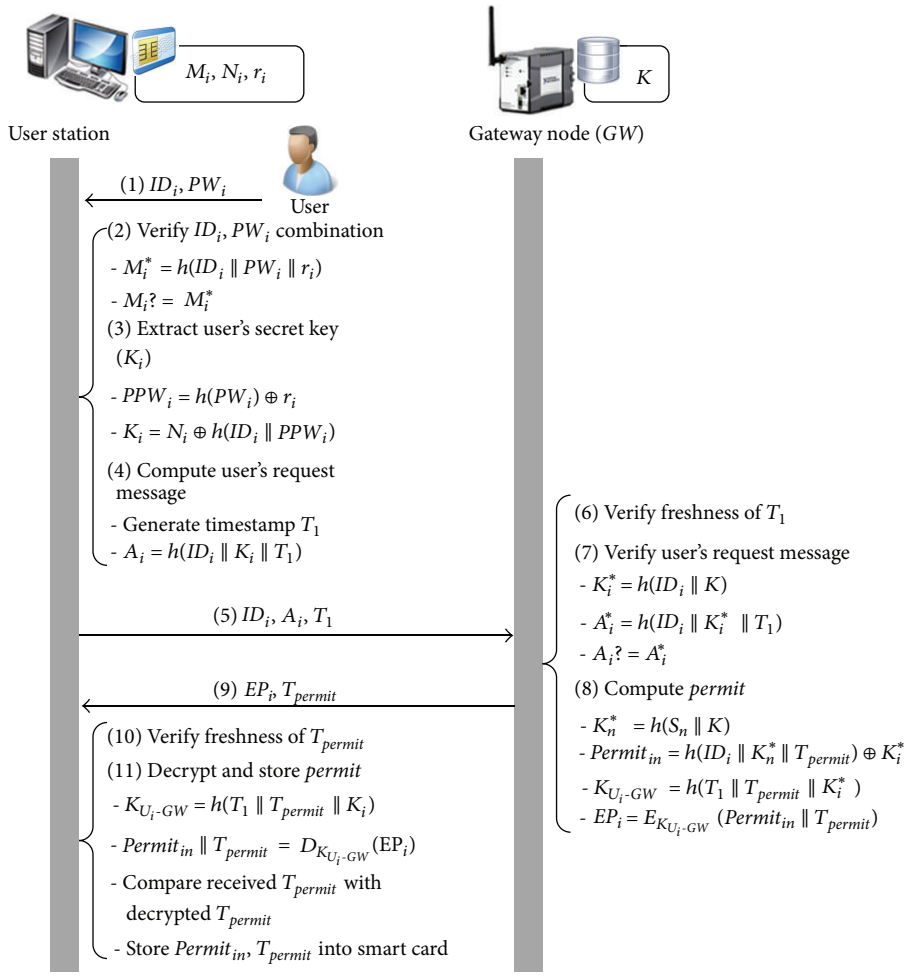


Figure 9: Proposed offline permit issue subprotocol.

if an encryption channel was required after authentication. Additionally, if a direct communication channel between U_i and S_n was required, a bilateral session key $K_{U_i\text{-}S_n}$ could be established through GW. In this case, GW would generate a random $K_{U_i\text{-}S_n}$ and send $T_5 \| K_{U_i\text{-}S_n}$ encrypted with $K_{U_i\text{-}GW}$ to U_i and $T_5 \| K_{U_i\text{-}S_n}$ encrypted with $K_{S_n\text{-}GW}$ to S_n, where T_5 is the timestamp of GW when sending the session key.

Note. Although the process of the online user authentication protocol is similar to that proposed by the existing works, it includes several enhancements. First, it delivers mutual authentication among all entities (U_i, GW, and S_n). Additionally, the proposed protocol includes a simple session key establishment phase which was not provided in most of solutions.

4.2.2. Offline or Gateway-Less User Authentication Protocol. In this use case, S_n is disconnected from the gateway node; therefore, it is not possible to receive the authorization from GW to provide access to the collected data to U_i (see Figure 6). In this situation, another form of authentication is required, where S_n must be sure that the U_i is an authenticated user authorized by GW without dealing directly with U_i's key; on the other hand, U_i also must be sure that S_n is an authentic sensor node, but without dealing with S_n's key. On that point, a special data called *Permit* is proposed to be used instead of the secret keys. *Permit* contains the authorization of GW to access the data of a sensor node.

The offline or gateway-less user authentication protocol is composed of two subprotocols, namely, permit issue and User Authentication. The first one allows U_i to request the permit for accessing S_n ($Permit_{in}$) to the gateway node, while the second one allows U_i to use the issued permit to authenticate to S_n.

(A) Permit Issue Subprotocol. After receiving the smart card from GW, U_i can use it to authenticate himself or herself to GW and receive the $Permit_{in}$ which will allow U_i to authenticate to the sensor node S_n when it is disconnected from GW. $Permit_{in}$ is a secret value issued by GW, which certifies that U_i has permission to access the data collected by S_n. The permit issue subprotocol is executed as follows (see Figure 9 and Table 2).

U_i inserts the smart card and inputs his/her ID_i and PW_i. The smart card then computes $M_i^* = h(ID_i \| PW_i \|$

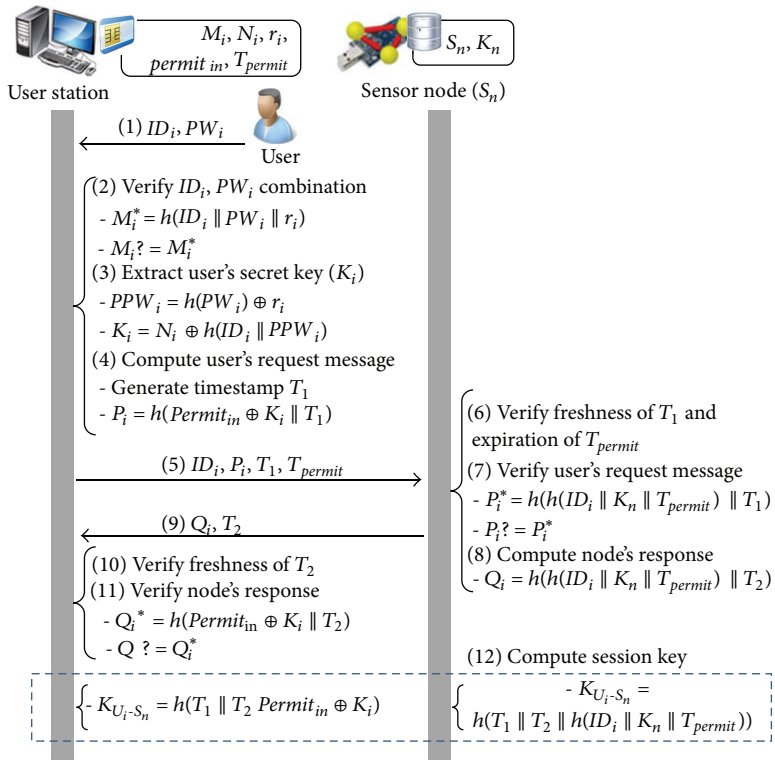


Figure 10: Proposed offline user authentication subprotocol.

r_i) and compares M_i^* with M_i to authenticate U_i. If those values do not match, the authentication request is rejected. Otherwise, the smart card computes $PPW_i = h(PW_i) \oplus r_i$, $K_i = N_i \oplus h(ID_i \| PPW_i)$, and $A_i = h(ID_i \| K_i \| T_1)$ and transmits $\{ID_i, A_i, T_1\}$ to GW, where T_1 is the current timestamp of U_i's system. Upon receiving the login request at time T_1^*, GW validates T_1. If $(T_1^* - T_1) > \Delta T$, GW aborts the authentication process, where ΔT denotes the maximum allowed communication delay. Otherwise, GW computes $K_i^* = h(ID_i \| K)$ and $A_i^* = h(ID_i \| K_i^* \| T_1)$. Once A_i^* is computed, GW checks whether it is equal to A_i. If A_i^* is different to A_i, GW finishes the permit issue process; otherwise, GW computes $K_n^* = h(S_n \| K)$, $Permit_{in} = h(ID_i \| K_n^* \| T_{permit}) \oplus K_i^*$, $K_{U_i\text{-}GW} = h(T_1 \| T_{permit} \| K_i^*)$, and encrypted permit $EP_i = E_{K_{U_i\text{-}GW}}(Permit_{in} \| T_{permit})$ and transmits $\{EP_i, T_{permit}\}$ to U_i, where T_{permit} is the timestamp of GW's system when generating $Permit_{in}$ and $E_{secret\_key}(string)$ is a symmetric encryption of a $string$ using the key $secret\_key$. U_i first validates T_{permit} using similar method of T_1 verification then computes $K_{U_i\text{-}GW} = h(T_1 \| T_{permit} \| K_i)$ to obtain $Permit_{in} \| T_{permit} = D_{K_{U_i\text{-}GW}}(EP_i)$, where $D_{secret\_key}(string)$ is a symmetric decryption of a $string$ using the key $secret\_key$. Once T_{permit} is obtained, U_i compares it with the T_{permit} received in plaintext to validate once more the validity of EP_i. Finally, U_i stores $Permit_{in}$ and T_{permit} into the smart card.

Note. The permit can be obtained (1) immediately after receiving the smart card as well as (2) when the need arises. The criterion for checking the validity of ΔT_{permit} is the same for both cases, that is, a period of time (e.g., one day, one week, etc.). The first case provides the advantage that the user can mobilize to a location near sensor nodes but without network connection with the GW. However, in this case, the user must be prudent in updating the *Permit* before its expiration. The second case can be used when the user stays in a location with stable network connection with the GW. We believe that obtaining the *permit* from the GW when the need arises is not a critical problem because the time required in executing the permit issue sub-protocol is small. In conclusion, taking into account the previous analysis, the first case (obtaining the *permit* immediately) could be a better solution, if the response time to the events was critical and the user was not in a place connected to the GW. On the other hand, the second case (obtaining the *permit* from the GW when the need arises) could be better when the user is located in a place with stable network connection with the GW.

(B) User Authentication Subprotocol. Once the *Permit* received, U_i can go to the field to authenticate directly to the sensor node. This protocol is composed of two phases, namely authentication and session key establishment (see Figure 10 and Table 2).

Authentication Phase. Once the $Permit_{in}$ is issued from GW, U_i can use it to authenticate himself or herself to S_n and access S_n's data. The authentication process is executed as follows. U_i inserts the smart card and inputs his/her ID_i and PW_i. The smart card then computes $M_i^* = h(ID_i \| PW_i \|$

r_i) and compares M_i^* with M_i to authenticate U_i. If those values do not match, the authentication request is rejected. Otherwise, the smart card computes $PPW_i = h(PW_i) \oplus r_i$, $K_i = N_i \oplus h(ID_i \parallel PPW_i)$, and $P_i = h(Permit_{in} \oplus K_i \parallel T_1)$ and transmits $\{ID_i, P_i, T_1, T_{permit}\}$ to S_n, where T_1 is the current timestamp of U_i's system. Upon receiving the login request at time T_1^*, S_n validates T_1 and T_{permit}. If $(T_1^* - T_1) > \Delta T$ or $(T_1^* - T_{permit}) > \Delta T_{permit}$, S_n aborts the authentication process, where ΔT denotes the maximum allowed communication delay and ΔT_{permit} denotes the period of validity of a permit. Otherwise, S_n computes $P_i^* = h(h(ID_i \parallel K_n \parallel T_{permit}) \parallel T_1)$. Once P_i^* is computed, S_n checks whether it is equal to P_i. If P_i^* is different to P_i, S_n finishes the authentication process; otherwise, S_n computes $Q_i = h(h(ID_i \parallel K_n \parallel T_{permit}) \parallel T_2)$ and transmits $\{Q_i, T_2\}$ to U_i, where T_2 is the actual timestamp of S_n's system. U_i first validates T_2 using similar method of T_1 verification then checks the validity of Q_i by comparing $Q_i^* = h(Permit_{in} \oplus K_i \parallel T_2)$ with the received D_i. If those values match, then U_i trusts S_n and accesses its data.

Session Key Establishment. A session key between U_i and S_n $K_{U_i\text{-}S_n} = h(T_1 \parallel T_2 \parallel h(ID_i \parallel K_n \parallel T_{permit}))$ could be used if an encryption channel was required after authentication.

4.3. Password Change Protocol. One of the requirements of a secure protocol is the delivery of a mechanism to users so that they can freely change their passwords. The proposed scheme provides a simple and efficient password change protocol which does not require communication with the gateway node. The proposed password change protocol is executed as follows (see Figure 11 and Table 2).

U_i inputs his/her ID_i, PW_i, and new password $NewPW_i$ to the smart card. The smart card then calculates $M_i^* = h(ID_i \parallel PW_i \parallel r_i)$ and verifies the validity of ID_i and PW_i by comparing M_i^* with M_i. If those values do not match, the password change request is rejected. Otherwise, the smart card computes $PPW_i = h(PW_i) \oplus r_i$, $h(ID_i \parallel K) = N_i \oplus h(ID_i \parallel PPW_i)$, $NewM_i = h(ID_i \parallel NewPW_i \parallel r_i)$, and $NewN_i = h(ID_i \parallel NewPPW_i) \oplus h(ID_i \parallel K)$, where $NewPPW_i = h(NewPW_i) \oplus r_i$. Finally, the smart card replaces M_i and N_i with $NewM_i$ and $NewN_i$, respectively.

4.4. Implementation Issue. This paper does not consider the details related to the real implementation of the proposed scheme. However, we would like to share several ideas required on implementing the proposed scheme. First, the user interface of the application to be installed in the user station (U_i) managing the execution of protocols must include the option to choose the protocol to be executed by the user. Second, as the content of the messages sent from U_i to GW in online user authentication protocol and offline permit issue subprotocol is the same ($\{ID_i, A_i, T_1\}$), it must include a header indicating which protocol GW must execute; one possible solution could be the usage of 1 bit: "0" for online user authentication and "1" for offline permit issue sub-protocol. Finally, the requests $\{ID_i, B_i, T_2\}$ and $\{ID_i, P_i, T_1, T_{permit}\}$ used in online and offline user authentication protocols are differentiated by S_n using messages' sizes. This means that S_n would execute the steps of one protocol according to the size of the receiving message. As another option, request messages could include a header to indicate the type of the request the user is asking for. The mentioned ideas are just possible options and we believe that the details required for real implementation should be considered in the implementation based on the requirements of the total solution.

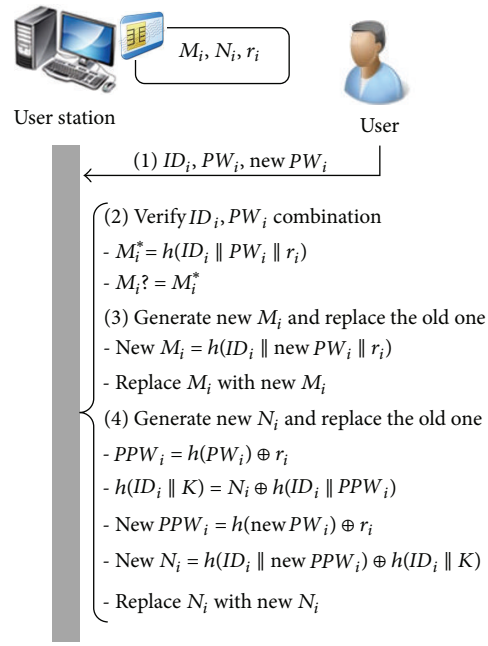


Figure 11: Proposed password change protocol.

5. Analysis of the Proposed Scheme

5.1. Security Analysis. This part analyzes the security of the proposed scheme in terms of formal verification and analysis of security requirements described in Section 3. The registration and password change protocols of the proposed scheme are excluded from this analysis because they are executed in a secure environment. In the analysis of the user authentication protocol, the threat model discussed in Section 3 is applied.

5.1.1. Formal Proof Based on BAN Logic

(A) Notations and Rules of BAN Logic. This subsection demonstrates the security of the proposed scheme by a well-known formal model called BAN logic [23, 24]. BAN logic has been widely used in different works such as [25–27] to reason about their security validation. The logical notations of BAN logic used in this paper are as described in Table 3. This section also lists some main logical postulates to be used in proofs.

Message-Meaning Rule. If the principal P believes that the secret key is shared with the principal Q and P sees that the statement X is encrypted or combined (hashed) under K then

Table 3: Notations of BAN logic.

Notation	Description
$P \mid\equiv X$	The principal P believes that X holds. In other words, it means that P is entitled to act as though X is true
$\#(X)$	The formula X is fresh. That is, X has not been sent before in any run of the protocol
$P \Rightarrow X$	The principal P has jurisdiction over the statement X. That is, P is an authority on X and can be trusted on X
$P \lhd X$	The principal P sees the statement X. That is, someone has sent a message to P containing X, and P can read and repeat X
$P \mid\sim X$	The principal P once said the statement X. That is, P sent a message containing X sometime
(X, Y)	The formula X or Y is one part of the formula (X, Y)
$\{X\}_K$	The formula X is encrypted under the key K
$(X)_K$	The formula X is hashed with the key K, and K may be used to prove the origin of X
$P \overset{k}{\leftrightarrow} Q$	Principals P and Q may use the shared key K to communicate. The key K will never be discovered by any principal except P and Q

the principal P believes that the principal Q once said the statement X:

$$\frac{P \mid\equiv P \overset{K}{\longleftrightarrow} Q, P \lhd \{X\}_K}{P \mid\equiv Q \mid\sim X}, \qquad \frac{P \mid\equiv P \overset{K}{\longleftrightarrow} Q, P \lhd (X)_K}{P \mid\equiv Q \mid\sim X}. \tag{1}$$

Freshness-Conjuncatenation Rule. Provided that the principal P believes freshness of the statement X, the principal P believes freshness of the (X, Y):

$$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X, Y)}. \tag{2}$$

Nonce-Verification Rule. Provided that the principal P believes that the statement X have never been utter before and the principal Q once said X, the principal P believes that Q believes X:

$$\frac{P \mid\equiv \#(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}. \tag{3}$$

Jurisdiction Rule. Provided that the principal P believes that the principal Q jurisdiction over the statement X, the principal P believes Q on the validity of X:

$$\frac{P \mid\equiv Q \Longrightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}. \tag{4}$$

Belief Rules. A necessary property of the belief operator is that P believes a set of statements if and only if P believes each statement separately. This justifies the following rules:

$$\frac{P \mid\equiv X, P \mid\equiv Y}{P \mid\equiv (X, Y)}, \qquad \frac{P \mid\equiv (X, Y)}{P \mid\equiv X},$$
$$\frac{P \mid\equiv Q \mid\equiv (X, Y)}{P \mid\equiv Q \mid\equiv X}, \qquad \frac{P \mid\equiv Q \mid\sim (X, Y)}{P \mid\equiv Q \mid\sim X}. \tag{5}$$

(B) Formal Proof. In the following, it shows the security proof of the authentication protocol using the BAN logic.

(1) Formal Proof of Online User Authentication. This part presents the formal proof of the online user authentication protocol, when sensor nodes have active communication link with the gateway node.

In this case, the protocol must satisfy the following goals:

$$GW \mid\equiv A_i, \tag{G.1}$$
$$S_n \mid\equiv B_i, \tag{G.2}$$
$$GW \mid\equiv C_i, \tag{G.3}$$
$$U_i \mid\equiv D_i, \tag{G.4}$$
$$U_i \mid\equiv K_{U_i\text{-}GW}, \tag{G.5}$$
$$GW \mid\equiv K_{U_i\text{-}GW}, \tag{G.6}$$
$$GW \mid\equiv K_{S_n\text{-}GW}, \tag{G.7}$$
$$S_n \mid\equiv K_{S_n\text{-}GW}. \tag{G.8}$$

First, messages of the protocol are transformed to the idealized form as shown below:

$$U_i \longrightarrow GW: \; ID_i, (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i \parallel K)}, T_1, \tag{M.1}$$
$$S_n \longleftarrow GW: \; ID_i, (ID_i \parallel h(S_n \parallel K) \parallel T_2)_{h(S_n \parallel K)}, T_2, \tag{M.2}$$
$$S_n \longrightarrow GW: (ID_i \parallel h(S_n \parallel K) T_3)_{h(S_n \parallel K)}, T_3, \tag{M.3}$$
$$U_i \longleftarrow GW: \; (ID_i \parallel h(ID_i \parallel K) \parallel T_4)_{h(ID_i \parallel K)}, T_4. \tag{M.4}$$

Second, assumptions about the initial state of the protocol are defined. Those assumptions are listed below:

$$GW \mid\equiv \#(T_1), \tag{A.1}$$
$$S_n \mid\equiv \#(T_2), \tag{A.2}$$
$$GW \mid\equiv \#(T_3), \tag{A.3}$$
$$U_i \mid\equiv \#(T_4), \tag{A.4}$$
$$U_i \mid\equiv U_i \overset{h(ID_i \parallel K)}{\longleftrightarrow} GW, \tag{A.5}$$
$$GW \mid\equiv U_i \overset{h(ID_i \parallel K)}{\longleftrightarrow} GW, \tag{A.6}$$
$$S_n \mid\equiv S_n \overset{h(S_n \parallel K)}{\longleftrightarrow} GW, \tag{A.7}$$
$$GW \mid\equiv S_n \overset{h(S_n \parallel K)}{\longleftrightarrow} GW, \tag{A.8}$$
$$GW \mid\equiv U_i \Longrightarrow A_i, \tag{A.9}$$
$$S_n \mid\equiv GW \Longrightarrow B_i, \tag{A.10}$$
$$GW \mid\equiv S_n \Longrightarrow C_i, \tag{A.11}$$
$$U_i \mid\equiv GW \Longrightarrow D_i. \tag{A.12}$$

Finally, the proof steps to the idealized form of the proposed protocol are performed based on BAN logic rules and assumptions. The proof steps are as in Table 21.

The proposed goals were reached by (S.5), (S.10), (S.15), (S.20), (S.21), (S.22), (S.23), and (S.24). In summary, this paper has demonstrated how the proposed online user authentication protocol is secure and provides mutual authentication among U_i, GW, and S_n.

(2) Formal Proof of Offline or Gateway-Less User Authentication

(a) Permit Issue Subprotocol. In the following, there is the security proof of the proposed permit issue sub-protocol. In this sub-protocol, goals to be reached are as follows:

$$GW \mid\equiv A_i, \quad \text{(G.1)}$$
$$U_i \mid\equiv Permit_{in}. \quad \text{(G.2)}$$

First, messages of the protocol are transformed to the idealized form as shown below:

$$U_i \longrightarrow GW: \quad ID_i, \left(ID_i \parallel h\left(ID_i \parallel K\right) \parallel T_1\right)_{h(ID_i \parallel K)}, T_1, \quad \text{(M.1)}$$
$$S_n \longleftarrow GW: \{Permit_{in}, T_{permit}\}_{K_{U_i\text{-}GW}}, T_{permit}. \quad \text{(M.2)}$$

Second, the following assumptions are made about the initial state of the sub-protocol to analyze the proposed scheme:

$$GW \mid\equiv \#\left(T_1\right), \quad \text{(A.1)}$$
$$U_i \mid\equiv \#(T_{permit}), \quad \text{(A.2)}$$
$$U_i \mid\equiv U_i \overset{h(ID_i \parallel K)}{\longleftrightarrow} GW, \quad \text{(A.3)}$$
$$GW \mid\equiv U_i \overset{h(ID_i \parallel K)}{\longleftrightarrow} GW, \quad \text{(A.4)}$$
$$GW \mid\equiv U_i \Longrightarrow A_i, \quad \text{(A.5)}$$
$$U_i \mid\equiv GW \Longrightarrow Permit_{in}. \quad \text{(A.6)}$$

Finally, proof steps to the idealized form of the proposed sub-protocol are performed based on the BAN logic rules and the assumptions as in Table 22.

The proposed goals were reached by (S.5) and (S.10). In summary, this paper has demonstrated how the proposed permit issue sub-protocol is secure and provides mutual authentication between U_i and GW.

(b) User Authentication Subprotocol. In the following, there is the security proof of the proposed user authentication sub-protocol using the BAN logic. The sub-protocol will satisfy the following goals:

$$S_n \mid\equiv P_i, \quad \text{(G.1)}$$
$$U_i \mid\equiv Q_i, \quad \text{(G.2)}$$
$$S_n \mid\equiv K_{U_i\text{-}S_n}, \quad \text{(G.3)}$$
$$U_i \mid\equiv K_{U_i\text{-}S_n}. \quad \text{(G.4)}$$

First, here is the transformation of protocol's messages to the idealized form:

$$U_i \longrightarrow S_n: \quad ID_i, \left(ID_i \parallel h\left(ID_i \parallel K_n \parallel T_{permit}\right) \parallel T_1\right)_{h(ID_i \parallel K_n \parallel T_{permit})}, \quad T_1, T_{permit}, \quad \text{(M.1)}$$
$$U_i \longleftarrow S_n: \quad \left(ID_i \parallel h\left(ID_i \parallel K_n \parallel T_{permit}\right) \parallel T_2\right)_{h(ID_i \parallel K_n \parallel T_{permit}), T_2}. \quad \text{(M.2)}$$

Second, the following assumptions are made about the initial state of the scheme to analyze the proposed scheme:

$$S_n \mid\equiv \#\left(T_1\right), \quad \text{(A.1)}$$
$$U_i \mid\equiv \#\left(T_2\right), \quad \text{(A.2)}$$
$$U_i \mid\equiv U_i \overset{h\left(ID_i \parallel K_n \parallel T_{permit}\right)}{\longleftrightarrow} S_n, \quad \text{(A.3)}$$
$$S_n \mid\equiv U_i \overset{h\left(ID_i \parallel K_n \parallel T_{permit}\right)}{\longleftrightarrow} S_n, \quad \text{(A.4)}$$
$$S_n \mid\equiv U_i \Longrightarrow P_i, \quad \text{(A.5)}$$
$$U_i \mid\equiv S_n \Longrightarrow Q_i. \quad \text{(A.6)}$$

Finally, the proof steps to the idealized form of the proposed sub-protocol are performed based on the BAN logic rules and the assumptions as in Table 23.

The proposed goals were reached by (S.5), (S.10), (S.11), and (S.12). In summary, this paper has demonstrated how the proposed user authentication sub-protocol is secure and provides mutual authentication between U_i and S_n.

5.1.2. Security Verification. This subsection analyzes the security of the proposed solution from the point of view of basic security requirements mentioned in Section 3. This section also analyzes how the proposed solution is secure against possible attacks. As mentioned previously, the paper assumes that the communication channels used in user authentication are insecure and that there exists an attacker who can intercept all messages exchanged among U_i, GW, and S_n. In addition, the paper assumes that the attacker can obtain or steal legal user U_i's smart card. Based on these assumptions, the attacker might execute certain attacks to interfere with the proposed scheme.

(A) Security Analysis of Basic Requirements

Confidentiality. Confidentiality of messages is guaranteed by usage of secret values. Communication between U_i and GW is encrypted using $K_i = h(ID_i \parallel K)$ while communication between GW and S_n is encrypted using $K_n = h(S_n \parallel K)$. Additionally, the communication messages in offline user authentication are encrypted by using a value generated from $Permit_{in}$. Furthermore, session keys ($K_{U_i\text{-}GW}$, $K_{S_n\text{-}GW}$, and $K_{U_i\text{-}S_n}$) are generated to guarantee the transmission of confidential values after authentication.

Integrity. Integrity of messages transmitted during the authentication process is guaranteed by verification in each step of protocols. In each verification step, entities, that is, U_i, GW, and S_n, compare the received data with values autonomously calculated by them to confirm the authenticity of messages.

Freshness. In network security protocols, the freshness and uniqueness of messages allow the provision of a strong defense against replay attacks. These two properties are reachable by using a time variant parameter inside messages, such

as random numbers, sequence numbers, and timestamps, being most common the usage of random numbers (nonces) and timestamps. This work has used the timestamp method because it is always more efficient in terms of number of communication rounds compared to the nonce-based counterpart [28].

(B) Security Verification from Possible Attacks

Privileged-Insider Attack. In the proposed solution, U_i transmits his/her pseudo-password $PPW_i = h(PW_i) \oplus r_i$ instead of PW_i. Therefore, GW will never know the PW_i value. This means that only U_i will know his/her secret password, protecting U_i in this way from a privileged-insider attack. Additionally, a random value r_i is incorporated inside PPW_i to make the discovery of PW_i harder.

Stolen-Verifier Attack. One of the features of the proposed protocol is the absence of a password/verifier table in GW and S_n. This feature prevents our solution from stolen-verifier attacks.

Replay Attack. Timestamps are used to avoid replay attacks. In each message, a different timestamp is used to guarantee its freshness.

Parallel Session Attack. In previous works, the attacker can obtain a valid authentication request message for the next timestamp because they use the XOR operation in a vulnerable way (see reference [8] for more details). In the proposed protocol, we have eliminated the possibility of parallel session attack by sending hash values instead of values resulting from XOR operations. Therefore, even though another legal user of the system eavesdrops on U_i's message $\{ID_i, A_i, T_1\}$ or $\{ID_i, P_i, T_1, T_{permit}\}$, he/she cannot obtain the secret values A_i or P_i for the next timestamp T_2 because they are irreversible one-way hash values.

Guessing Attack. In the proposed scheme, secret values are never sent in plaintext but encrypted by a one-way hash function or symmetric cryptography algorithm. Therefore, even when the adversary got A_i, B_i, C_i, D_i, EP_i, P_i, or Q_i, he or she could not guess any secret value (i.e., PW_i, K_i, K_n, or K) because of the secure property of the hash and symmetric cryptography algorithms.

Gateway-Node Bypassing Attack. The reason for the possibility of a GW bypassing attack in [3, 6] is due to the sharing of secret parameter x_a with S_n and U_i. If the value of x_a is compromised, then the whole sensor network will become vulnerable to the gateway node bypassing attack. On the other hand, the reason for the possibility of the gateway node bypassing attack in [7] is due to the secret value x_s stored in sensor nodes which can be extracted using similar method of extracting x_a from a smart card [11–13]; if x_s is extracted, the adversary can execute the GW bypassing attack using $DID_f = h(ID_f \parallel PW_f) \oplus h(x_a \parallel T_f)$ and $A_f = h(DID_f \parallel S_n \parallel x_s \parallel T_f)$.

In the proposed protocol, U_i's smart card and S_n do not store either x_a or x_s but instead store other individual secret values $N_i = h(ID_i \parallel PPW_i) \oplus (ID_i \parallel K)$ and $K_n = h(S_n \parallel K)$ which are unique per smart card and sensor node. Therefore, even if the K_n value was extracted from a sensor node, the rest of nodes will still maintain their security. On the other hand, even if the N_i were extracted from a smart card, the N_i would be unusable without the correct PW_i value because the A_i used for authentication is generated by using the $h(ID_i \parallel K)$ which can only be obtained with the correct $h(ID_i \parallel PPW_i)$ value.

User Impersonation. An adversary who wants to impersonate a valid user U_i to log into the network must calculate a valid A_i (for the online and offline user authentication) or P_i (for the permit issue). Since $A_i = h(ID_i \parallel K_i \parallel T_1)$ and $P_i = h(Permit_{in} \oplus K_i \parallel T_1)$ are calculated by a one-way hash function, the adversary cannot decipher such values. Additionally, A_i and P_i cannot be created arbitrarily because they are based on the secret value K_i obtainable only with the correct N_i and PW_i.

Gateway-Node Impersonation. An adversary who wants to impersonate a valid GW must calculate a valid B_i (for the online user authentication) or EP_i (for the permit issue). Since $B_i = h(ID_i \parallel K_n \parallel T_2)$ and $EP_i = E_{K_{U_i\text{-}GW}}(Permit_{in} \parallel T_{permit})$ are calculated using one-way hash function or a secure symmetric cryptographic algorithm, the adversary cannot decipher such values. Additionally, B_i and EP_i cannot be created arbitrarily because they are based on the secret value K which is only known by the authentic GW.

Sensor Node Impersonation. An adversary who wants to impersonate a valid S_n must calculate a valid C_i (for the online user authentication) or Q_i (for the offline user authentication). Since $C_i = h(ID_i \parallel K_n \parallel T_3)$ and $Q_i = h(h(ID_i \parallel K_n \parallel T_{permit}) \parallel T_2)$ are calculated using one-way hash function, the adversary cannot decipher such values. Additionally, C_i and Q_i cannot be created arbitrarily because they are based on the secret value K_n which is only known by the authentic GW.

Many Logged-In Users with the Same Login-ID. By using two-factor based authentication, the proposed scheme offers higher protection than only password-based schemes. Assuming that the U_i's smart card is not cloned, the proposed protocol successfully prevents this threat because the authentication process requires computation executed inside the valid smart card.

Brute-Force Attack. An attacker can try two kinds of brute-force attacks. (1) First, the attacker can attempt to authenticate by sending random or sequential messages (A_i, B_i, or P_i) to GW or S_n. However, as well as explained in the replay attack, this attack becomes infeasible because each authentication process uses a different timestamp. (2) On the other hand, an insider with a valid smart card can try to discover the secret values by performing brute-force attacks. However, the determination of those values is infeasible because they are stored using secure one-way hash functions. If higher level of protection for K was required, additional random numbers R_i and R_n could be added for the generation of $K_i = h(ID_i \parallel K \parallel R_i)$ and $K_n = h(S_n \parallel K \parallel R_n)$,

Table 4: List of enhanced security features of the proposed scheme.

Security feature	Proposed	Das's [3]	Nyang-Lee's [4]	Huang et al.'s [5]	Chen-Shih's [6]	Khan-Alghathbar's [7]
Secure password change/update	Yes	No	No	Yes	No	Yes
Protection against insider's attack	Yes	No	No	No	No	Yes
Protection against *GW* bypassing attack	Yes	No	No	No	No	Partial
Protection against parallel session attack	Yes	No	Yes	No	No	No
Mutual authentication between *GW* and S_n	Yes	No	Yes	No	No	Yes
Mutual authentication between U_i and *GW*	Yes	No	Yes	No	No	No
Session key establishment	Yes	No	Yes	No	No	No
Offline user authentication	Yes	No	No	No	No	No

respectively, which would be stored secretly in the *GW*. By using those additional random numbers, the number of possible combinations to decipher K_i and K_n is increased by 2^n times, where n is the size in bits of R_i and R_n.

(C) Security Verification of Other Security Requirements

Mutual Authentication. The proposed protocol provides both mutual authentication between U_i and *GW* as well as between *GW* and S_n,

(i) *Online User Authentication.* (1) The mutual authentication between U_i and *GW* is verified as follows. *GW* verifies the authenticity of U_i by comparing A_i sent by U_i with the A_i^* value calculated by itself. A_i can only be computed by the authentic U_i because it is based on the secret value $h(ID_i \parallel K)$ only calculable with the valid N_i and PW_i which are personal to each U_i. In the same way, U_i verifies the authenticity of *GW* by comparing D_i sent by *GW* with the D_i^* value computed by U_i. D_i can only be computed by the authentic *GW* because it is based on the secret value K only known by the authentic *GW*. (2) On the other hand, the mutual authentication between *GW* and S_n is verified as follows. S_n verifies the authenticity of *GW* by comparing B_i sent by *GW* with the B_i^* value calculated by itself. B_i can only be computed by the authentic *GW* because it is based on the secret value K. In the same way, *GW* verifies the authenticity of S_n by comparing C_i sent by *GW* with the C_i^* value computed by *GW*. C_i can only be computed by the authentic S_n because it is based on the secret K_n value only known by the specific S_n.

(ii) *Offline or Gateway-Less User Authentication/Permit Issue Subprotocol.* The mutual authentication between U_i and *GW* is verified as follows. *GW* verifies the authenticity of U_i by comparing A_i sent by U_i with the A_i^* value calculated by itself. A_i can only be computed by the authentic U_i because it is based on the secret value $h(ID_i \parallel K)$ only calculable with the valid N_i and PW_i which are personal to each U_i. On the other hand, U_i verifies the authenticity of *GW* by comparing T_{permit} sent in plaintext by *GW* with the T_{permit} value contained inside EP_i. EP_i can only be computed by the authentic *GW* because it is based on the secret value K only known by *GW*.

(iii) *Offline or Gateway-Less User Authentication/User Authentication Subprotocol.* The mutual Authentication between U_i and S_n is verified as follows. S_n verifies the authenticity of U_i by comparing P_i sent by U_i with the P_i^* value calculated by itself. P_i can only be computed by the authentic U_i because it is based on the secret value $Permit_{in} \oplus K_i$ only calculable by the authentic U_i. On the other hand, U_i verifies the authenticity of S_n by comparing Q_i sent by S_n with the Q_i^* value calculated by U_i. Q_i can only be computed by the authentic S_n because it is based on the secret value K_n only known by S_n.

Password Change Phase. Our proposal offers a lightweight password change phase that does not require communication with *GW*, making it secure and efficient.

Session Key Establishment. Our proposal offers a simple and practical method for session key establishment among U_i, *GW*, and S_n.

(D) Summary of Security Features of the Proposed Solution. Table 4 shows the comparison of security features among different works. This demonstrates how our scheme is stronger in terms of security. Our approach provides protection against different kinds of attacks (privileged-insider attack, gateway-node bypassing attack), also provides a secure password change phase and session key establishment, and achieves complete mutual authentication (mutual authentication between *GW* and S_n and between U_i and *GW*) features that previous works do not offer or offer with limitations.

5.2. Performance and Cost Analysis

5.2.1. Cryptographic Operations. Table 5 indicates the number of cryptographic operations required in each protocol per entity for online user authentication. It shows that our protocol requires a few more operations in the verification phase than some previous works. However, the majority of additional operations are executed by U_i or *GW* infrastructure which has no energy or computation power limitations. Therefore, the additional operations are not an impediment for real implementation. Additionally, it is possible to say that the additional operations are justifiable considering that our scheme includes security features that previous works do not offer, which is indispensable for implementing a reliable

Table 5: Number of cryptographic operations in online user authentication.

Protocol	Entity	Proposed	Das' [3]	Nyang-Lee's [4]	Huang et al.'s [5]	Chen-Shih's [6]	Khan-Alghathbar's [7]
User registration protocol	U_i	2 h	0 h	0 h	1 h	0 h	1 h
	GW	2 h	3 h	3 h	3 h	3 h	2 h
	S_n	0 h	0 h	0 h	0 h	0 h	0 h
Password change protocol	U_i	7 h	n/a	n/a	4 h	n/a	4 h
	GW	0 h	n/a	n/a	0 h	n/a	0 h
	S_n	0 h	n/a	n/a	0 h	n/a	0 h
Online user authentication protocol	U_i	5 h	4 h	7 h + 1 sd	4 h	5 h	4 h
	GW	6 h	4 h	8 h + 1 se	6 h	5 h	5 h
	S_n	2 h	1 h	4 h + 1 se + 1 sd	1 h	1 h	2 h

h: hash, se: symmetric encryption, and sd: symmetric decryption.

Table 6: Number of cryptographic operations in offline user authentication.

Subprotocol	Entity	Number of operations
Permit issue subprotocol	U_i	5 h + 1 sd
	GW	5 h + 1 se
Authentication subprotocol	U_i	5 h
	S_n	3 h

h: hash, se: symmetric encryption, and sd: symmetric decryption.

and trustworthy network. It is important to remember that a failure at the component level will often compromise the security of the entire system [29]. Furthermore, the proposed solution provides further usability delivering offline authentication when sensor nodes cannot communicate with the gateway node which is really useful in different kinds of applications such as the military and natural phenomenon monitoring. Table 6 shows the number of cryptographic operations required to execute the offline user authentication protocol. The permit issue sub-protocol requires 5 hash executions and 1 symmetric cryptographic operation in both U_i and GW. However, we believe this overhead is not an impediment for a real implementation because U_i and GW are powerful entities without resource limitations. On the other hand, the additional hash execution required by S_n in the offline user authentication sub-protocol (compared to the proposed online user authentication) is not an impediment for real implementation because it is not executed all the time but only in extraordinary situations.

5.2.2. Number of Messages. Analyzing the number of messages transmitted and received by sensor nodes is considered important because it affects the energy consumption of those devices. This paper only focuses on the protocols where the resource limited sensor nodes participate in (i.e., online and offline user authentication protocols). Table 7 shows the number of transmissions and receptions executed by different entities in online user authentication protocols in different proposals. It illustrates how the proposed protocol maintains equal or less number of messages than previous works to show its competitiveness.

Going into details of Table 7, the present work also presents the content of messages transmitted and received by different entities and their sizes in each solution (see Tables 8, 9, 10, 11, 12, and 13). This work has considered that the size of the different data inside of messages, such as user identification, timestamps, hash values, and login confirmation messages, is equal because the intention of this part is to analyze the number of data units in each protocol (analysis using more realistic data sizes is executed in Section 5.2.3). As you can see in Tables 8–13, the most lightweight solutions in terms of messages sizes in sensor nodes are those proposed by Das, Huang et al., and Chen-Shih. However, messages sizes in those protocols are reduced because they omit steps required to provide mutual authentication between the gateway-node and sensor nodes opening serious vulnerabilities. Therefore, even though they offer less communication, they cannot be considered as optimal solutions. On the other hand, Table 9 shows how Nyang-Lee's proposal has the highest communication overhead requiring 4 data units for transmission and 4 data units for reception in sensor nodes. Finally, Alghathbar's proposal and the proposed solution have the same communication overhead in sensor nodes with 2 data units in transmission and 3 data units in reception.

On the other hand, Table 14 shows the number of messages in offline user authentication. In this protocol, only one message exchange between U_i and S_n is executed, showing how it is efficient in this aspect. In case of S_n, it only requires the transmission and reception of 2 and 4 data units, respectively.

5.2.3. Energy Consumption. One of the most sensible limitations in sensor nodes is their energy capacity. Therefore, it is important to analyze the energy consumption in those devices. The energy consumptions of user station and gateway node have not been analyzed because they do not suffer from this aspect. Following the fact that the battery power of a sensor node is depleted by computational processing and radio consumption [30], this paper has calculated the energy consumption overhead caused by (1) cryptographic

Table 7: Number of transmissions and receptions in online user authentication.

Entity	Proposed		Das' [3]		Nyang-Lee's [4]		Huang et al.'s [5]		Chen-Shih's [6]		Khan-Alghathbar's [7]	
	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx
U_i	1	1	1	1	1	1	1	1	1	2	1	1
GW	2	2	2	2	2	2	2	2	3	2	2	2
S_n	1	1	1	1	1	1	1	1	1	1	1	1

Tx: transmission; Rx: reception.

Table 8: Details of communication messages of Das' proposal [3].

Entity	Transmission (Tx)		Reception (Rx)	
	Content of messages	Total size	Content of messages	Total size
U_i	$\{DID_i, C_i, T\}$	3	$\{Login\ OK\}$	1
GW	$\{DID_i, A_i, T\}$, $\{Login\ OK\}$	4	$\{DID_i, C_i, T\}$, $\{Login\ OK\}$	4
S_n	$\{Login\ OK\}$	1	$\{DID_i, A_i, T'\}$	3

Table 9: Details of communication messages of Nyang-Lee's proposal [4].

Entity	Transmission (Tx)		Reception (Rx)	
	Content of messages	Total size	Content of messages	Total size
U_i	$\{DID_i, C_i, T\}$	3	$\{S_n, R, T'', B_i\}$	4
GW	$\{DID_i, D_i, T' A_i\}$, $\{S_n, R, T'', B_i\}$	8	$\{DID_i, C_i, T\}$, $\{S_n, R, T'', B_i\}$	7
S_n	$\{S_n, R, T'', B_i\}$	4	$\{DID_i, D_i, T', A_i\}$	4

Table 10: Details of communication messages of Huang et al.'s proposal [5].

Entity	Transmission (Tx)		Reception (Rx)	
	Content of messages	Total size	Content of messages	Total size
U_i	$\{DID_i, C_i, T\}$	3	$\{Login\ OK\}$	1
GW	$\{DID_i, A_i, T'\}$, $\{Login\ OK\}$	4	$\{DID_i, C_i, T\}$, $\{Login\ OK\}$	4
S_n	$\{Login\ OK\}$	1	$\{DID_i, A_i, T'\}$	3

Table 11: Details of communication messages of Chen-Shih's proposal [6].

Entity	Transmission (Tx)		Reception (Rx)	
	Content of messages	Total size	Content of messages	Total size
U_i	$\{DID_i, C_i, T_u, R_i\}$	4	$\{C_g, R_c, S_n\}$, $\{Login\ OK\}$	4
GW	$\{DID_i, A_i, T'\}$, $\{C_g, R_c, S_n\}$, $\{Login\ OK\}$	7	$\{DID_i, C_i, T_u, R_i\}$, $\{Login\ OK\}$	5
S_n	$\{Login\ OK\}$	1	$\{DID_i, A_i, T'\}$	3

Table 12: Details of communication messages of Khan-Alghathbar's proposal [7].

Entity	Transmission (Tx)		Reception (Rx)	
	Content of messages	Total size	Content of messages	Total size
U_i	$\{DID_i, C_i, T\}$	3	$\{Login\ OK\}$	1
GW	$\{DID_i, A_i, T'\}$, $\{Login\ OK\}$	4	$\{DID_i, C_i, T\}$, $\{B_i, T''\}$	5
S_n	$\{B_i, T''\}$	2	$\{DID_i, A_i, T'\}$	3

Table 13: Details of communication messages of the proposed scheme (online user authentication).

Entity	Transmission (Tx)		Reception (Rx)	
	Content of messages	Total size	Content of messages	Total size
U_i	$\{ID_i, A_i, T_1\}$	3	$\{D_i, T_4\}$	2
GW	$\{ID_i, B_i, T_2\}$, $\{D_i, T_4\}$	5	$\{ID_i, A_i, T_1\}$, $\{C_i, T_3\}$	5
S_n	$\{C_i, T_3\}$	2	$\{ID_i, B_i, T_2\}$	3

Table 14: Details of communication messages of the proposed scheme (offline user authentication).

Entity	Transmission (Tx)			Reception (Rx)		
	Number of messages	Content of messages	Total size	Number of messages	Content of messages	Total size
U_i	1	$\{ID_i, P_i, T_1, T_{permit}\}$	4	1	$\{Q_i, T_2\}$	2
S_n	1	$\{Q_i, T_2\}$	2	1	$\{ID_i, P_i, T_1, T_{permit}\}$	4

Table 15: Energy consumption of cryptographic operations executing different online user authentication protocols.

Scheme	Energy (mJ)	
	MIPS R4000	MC68328
Proposed	0.0041472	0.023616
Das' [3]	0.0032256	0.018368
Nyang-Lee's [4]	0.0167040	0.111040
Huang et al.'s [5]	0.0027648	0.015744
Chen-Shih's [6]	0.0032256	0.018368
Khan-Alghathbar's [7]	0.0052992	0.030176

operations and (2) radio communications in executing the proposed security mechanism.

(A) Energy Consumption Overhead of Cryptographic Operations. First, for the calculation of energy used by cryptographic operations, this paper has used the energy consumption estimates indicated in [16]. In [16], the authors estimated that the energy per bit consumed by MIPS R4400 and MC68328 "DragonBall" processors for performing AES encryption/decryption operations is 9 nJ/bit and 101 nJ/bit, respectively, while for the SHA-1 hashing function the same processors consume 7.2 nJ/bit and 41 nJ/bit, respectively. Additionally, this work has assumed that the size of random numbers is 160 bits, the sizes of timestamps, identifications (ID of users, gateway node, and sensor nodes), and Login OK messages are 64 bits each, and the sizes of secret values such as x_a, x_s, x_n, K_n are 160 bits each. Using those values, the energy consumption of cryptographic operation in each scheme was calculated (see Table 15). Here is an example of how the energy consumptions of cryptographic operation were calculated: in the proposed online authentication protocol, two hash values are calculated by S_n, that is, B_i^* and C_i. As the input values for the generation of B_i^* and C_i are 288 bits each, the energy used in generating B_i^* and C_i will be 7.2 nJ/bit $*$ 288 bits = 0.002074 mJ for MIPS R4400 processor and 41 nJ/bit $*$ 288 bits = 0.011808 mJ for MC68628 processor. Therefore, the total energy used by the cryptographic operations in the proposed online user authentication protocol will be the summation of the energy consumptions in generating B_i^* and C_i, that is, 0.0041472 mJ and 0.023616 mJ for MIPS R4400 and MC68628 processors, respectively. Table 15 shows how the proposed solution consumes 0.00092, 0.00138, and 0.00092 mJ more than M. Das, Huang et al., and Chen-Shih's schemes, respectively, and consumes 0.012557 and 0.001152 mJ less than Nyang-Lee and Khan-Alghathbar's schemes, respectively, when using the MIPS R4400 processor. Using the data of the table, it is also possible to deduce that the proposed solution consumes 0.002097 mJ less than the average energy consumption of the rest of approaches, which means it has competitive energy consumption compared to the rest of solutions. Similar conclusion can be reached with the energy consumption values when using the MC68328 microprocessor.

Table 16: Energy consumption of cryptographic operations executing the proposed offline user authentication sub-protocol.

Energy (mJ)	
MIPS R4000	MC68328
0.0052992	0.030176

Table 17: Radio communication parameters.

Parameter	Value
Transmitter/receiver Electronics E_{elec}	50 nJ/bit
Transmitter amplifier E_{amp}	10 pJ/bit
Distance d	50 m
Transmission energy model E_{TX}	$E_{Tx} = (E_{elec} + E_{amp} * d^2) * k$
Reception energy model E_{Rx}	$E_{Rx} = E_{elec} * k$

On the other hand, this work also has calculated the energy consumption of sensor nodes in executing cryptographic operations during the offline user authentication sub-protocol (see Table 16). It shows how this approach consumes 0.001152 and 0.00656 mJ more than the online user authentication using MIPS R4000 and MC68328 processors, respectively. However, we believe that it is not an impediment for real implementation because offline user authentication is not executed all the time but only in extraordinary situations where online user authentication cannot cover.

(B) Energy Consumption Overhead of Radio Communication. For the calculation of energy consumption of radio communication produced by the proposed protocols, the present work has assumed a simple model where the radio dissipates E_{elec} = 50 nJ/bit to run the transmitter or receiver circuitry and E_{amp} = 10 pJ/bit/m^2 to run the transmitter amplifier (see Table 17), similar to the model used in [31–33]. Additionally, for the calculation of the energy consumption, this work has used the parameters indicated in Table 17 and has assumed that the implemented hash function is SHA1 (160 bits hash value), the sizes of random numbers are 160 bits, the sizes of timestamps, identifications (ID of users, gateway node, and sensor nodes), and Login OK messages are 64 bits each, and the sizes of secret values such as x_a, x_s, x_n, K_n are 160 bits each.

Table 18 indicates the energy used by a sensor node in executing the online user authentication in different schemes.

Table 18: Energy consumption of radio communication in different protocols.

Scheme	Energy (mJ)
Proposed online/offline user authentication	0.0312/0.0344
Das' [3]	0.0240
Nyang-Lee's [4]	0.0568
Huang et al.'s [5]	0.0240
Chen-Shih's [6]	0.0240
Khan-Alghathbar's [7]	0.0360

Here we show an example of how such values were calculated. In the proposed online authentication protocol, the message $\{ID_i, B_i, T_2\}$ is received and the message $\{C_i, T_3\}$ is sent by S_n. In this example, the total radio communication energy consumption is calculated by adding the energy used in reception and transmission of messages. As the receiving message's size is 288 bits, the radio communication energy used for reception will be 50 nJ/bit $*$ 288 bits = 0.0144 mJ. On the other hand, as the transmitting message's size is 224 bits, the radio communication energy used for transmission will be (50 nJ + 10 pJ/bit/m^2 $*$ (50 m)^2) $*$ 224 bits = 0.0168 mJ. Then, the total radio communication energy used in the proposed online authentication protocol will be 0.0144 mJ + 0.0168 mJ = 0.0312 mJ. Table 18 shows how the proposed solution consumes 0.0072 mJ more than M. Das, Huang et al.'s and Chen-Shih's schemes and consumes 0.0256 and 0.0048 mJ less than Nyang-Lee's and Khan-Alghathbar's schemes, respectively. It also illustrates how the proposed solution consumes 0.000176 mJ less than the average energy consumption of the rest of approaches, which means it has competitive energy consumption compared to the rest of solutions.

On the other hand, the offline user authentication sub-protocol only consumes 0.0032 mJ more than the proposed online user authentication protocol. We believe that this difference is acceptable considering the extraordinary situation when the offline user authentication is used.

(C) Total Energy Consumption Overhead. The total energy consumption overhead in sensor nodes can be calculated adding the energy consumption of cryptographic operations and radio communication. Table 19 shows the total energy consumption overhead of different schemes. It shows that the proposed solution consumes 0.008122, 0.008582, and 0.008122 mJ more than M. Das's, Huang et al.'s, and Chen-Shih's schemes, respectively, and consumes 0.03816 and 0.00595 mJ less than Nyang-Lee's and Khan-Alghathbar's schemes, respectively, when using the MIPS R4400 processor. It also illustrates how the proposed solution consumes 0.00386 mJ less than the average energy consumption of the rest of approaches, which means it has competitive energy consumption compared to the rest of solutions. Similar conclusion can be reached with the energy consumption of schemes when using the MC68328 microprocessor. It is important to say that the additional energy overhead from other works is justifiable considering that our scheme includes security features that previous works do not offer, which is indispensable for implementing a reliable and trustworthy network.

Table 19: Total energy consumption overhead of authenticating sensor nodes executing different online user authentication protocols.

Scheme	Total energy (mJ)	
	MIPS R4000	MC68328
Proposed	0.0353472	0.054816
Das' [3]	0.0272256	0.042368
Nyang-Lee's [4]	0.0735040	0.167840
Huang et al.'s [5]	0.0267648	0.039744
Chen-Shih's [6]	0.0272256	0.042368
Khan-Alghathbar's [7]	0.0412992	0.066176

Table 20: Total energy consumption overhead of authenticating sensor nodes executing the proposed offline user authentication sub-protocol.

Total energy (mJ)	
MIPS R4000	MC68328
0.0396992	0.064576

On the other hand, Table 20 shows the total energy consumption of authenticating sensor nodes executing the proposed offline user authentication sub-protocol. It shows that this approach consumes 0.004352 and 0.00976 mJ more than the online user authentication using MIPS R4000 and MC68328 processors, respectively. However, we believe that it is not an impediment for real implementation because offline user authentication is not executed all the time but only in extraordinary situations where online user authentication cannot cover.

(D) Effect of Energy Consumption Overhead in the Wireless Sensor Network. Although there are differences among the energy consumption in different schemes, as they are small, it could not be neglected. Therefore, this paper has analyzed how much the energy consumption overhead of the proposed solution affects the lifetime of the sensor network. According to [16], one of the most common typical batteries in sensor nodes is the MN1500 Duracell AA with energy potential of 15.39 kJ. Based on this data, this work calculates how long a sensor node can survive executing the proposed user authentication scheme. In this simulation, we assume that the user authentication can use from 1% to 5% of the total energy while the rest of energy is used by other functionalities of the sensor node, such as path maintenance, data gathering, and data transmission.

Knowing that common sensor network applications are not dedicated for a massive user access yet, this work assumes that it is acceptable to use the average of one user authentication per minute as parameter. However, to understand how the proposed user authentication can act from higher demand, the present paper also considers the case when the average number of user authentication per minute is five. Figure 12 shows the number of months a sensor node with MIPS R4000 microprocessor can survive when executing 1 and 5 online user authentications per minute assuming that

Table 21

$GW \triangleleft (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i \parallel K)}$	(S.1)	// by (M.1)
$GW \mid\equiv U_i \mid\sim (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i \parallel K)}$	(S.2)	// by (S.1), (A.6), and message-meaning rule
$GW \mid\equiv U_i \mid\equiv (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i \parallel K)}$	(S.3)	// by (S.2), (A.1), freshness-conjuncatenation rule, and nonce-verification rule
$GW \mid\equiv (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i \parallel K)}$	(S.4)	// by (S.3), (A.9), and jurisdiction rule.
$GW \mid\equiv A_i$	(S.5)	// by (S.4) and as $A_i = (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i \parallel K)}$ (G.1)
$S_n \triangleleft (ID_i \parallel h(S_n \parallel K) \parallel T_2)_{h(S_n \parallel K)}$	(S.6)	// by (M.2)
$S_n \mid\equiv GW \mid\sim (ID_i \parallel h(S_n \parallel K) \parallel T_2)_{h(S_n \parallel K)}$	(S.7)	// by (S.6), (A.7), and message-meaning rule
$S_n \mid\equiv GW \mid\equiv (ID_i \parallel h(S_n \parallel K) \parallel T_2)_{h(S_n \parallel K)}$	(S.8)	// by (S.7), (A.2), freshness-conjuncatenation rule, and nonce-verification rule
$S_n \mid\equiv (ID_i \parallel h(S_n \parallel K) \parallel T_2)_{h(S_n \parallel K)}$	(S.9)	// by (S.8), (A.10), and jurisdiction rule.
$S_n \mid\equiv B_i$	(S.10)	// by (S.9) and as $B_i = (ID_i \parallel h(S_n \parallel K) \parallel T_2)_{h(S_n \parallel K)}$ (G.2)
$GW \triangleleft (ID_i \parallel h(S_n \parallel K) \parallel T_3)_{h(S_n \parallel K)}$	(S.11)	// by (M.3)
$GW \mid\equiv S_n \mid\sim (ID_i \parallel h(S_n \parallel K) \parallel T_3)_{h(S_n \parallel K)}$	(S.12)	// by (S.11), (A.8), and message-meaning rule
$GW \mid\equiv S_n \mid\equiv (ID_i \parallel h(S_n \parallel K) \parallel T_3)_{h(S_n \parallel K)}$	(S.13)	// by (S.12), (A.3), freshness-conjuncatenation rule, and nonce-verification rule
$GW \mid\equiv (ID_i \parallel h(S_n \parallel K) \parallel T_3)_{h(S_n \parallel K)}$	(S.14)	// by (S.13), (A.11), and jurisdiction rule.
$GW \mid\equiv C_i$	(S.15)	// by (S.14) and as $C_i = (ID_i \parallel h(S_n \parallel K) \parallel T_3)_{h(S_n \parallel K)}$ (G.3)
$U_i \triangleleft (ID_i \parallel h(ID_i \parallel K) \parallel T_4)_{h(ID_i \parallel K)}$	(S.16)	// by (M.4)
$U_i \mid\equiv GW \mid\sim (ID_i \parallel h(ID_i \parallel K) \parallel T_4)_{h(ID_i \parallel K)}$	(S.17)	// by (S.16), (A.5), and message-meaning rule
$U_i \mid\equiv GW \mid\equiv (ID_i \parallel h(ID_i \parallel K) \parallel T_4)_{h(ID_i \parallel K)}$	(S.18)	// by (S.17), (A.4), freshness-conjuncatenation rule, and nonce-verification rule
$U_i \mid\equiv (ID_i \parallel h(ID_i \parallel K) \parallel T_4)_{h(ID_i \parallel K)}$	(S.19)	// by (S.18), (A.12), and jurisdiction rule.
$U_i \mid\equiv D_i$	(S.20)	// by (S.19) and as $D_i = (ID_i \parallel h(ID_i \parallel K) \parallel T_4)_{h(ID_i \parallel K)}$ (G.4)
$U_i \mid\equiv \#(K_{U_i\text{-}GW}), U_i \mid\equiv K_{U_i\text{-}GW}$	(S.21)	// once (S.20) is verified, U_i calculates $K_{U_i\text{-}GW}$ (G.5)
$GW \mid\equiv \#(K_{U_i\text{-}GW}), GW \mid\equiv K_{U_i\text{-}GW}$	(S.22)	// once (S.5) is verified, GW calculates $K_{U_i\text{-}GW}$ (G.6)
$GW \mid\equiv \#(K_{S_n\text{-}GW}), GW \mid\equiv K_{S_n\text{-}GW}$	(S.23)	// once (S.15) is verified, GW calculates $K_{S_n\text{-}GW}$ (G.7)
$S_n \mid\equiv \#(K_{S_n\text{-}GW}), S_n \mid\equiv K_{S_n\text{-}GW}$	(S.24)	// once (S.10) is verified, S_n calculates $K_{S_n\text{-}GW}$ (G.8)

Table 22

$GW \triangleleft (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i \parallel K)}$	(S.1)	// by (M.1)
$GW \mid\equiv U_i \mid\sim (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i \parallel K)}$	(S.2)	// by (S.1), (A.4), and message-meaning rule
$GW \mid\equiv U_i \mid\equiv (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i \parallel K)}$	(S.3)	// by (S.2), (A.1), freshness-conjuncatenation rule, and nonce-verification rule
$GW \mid\equiv (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i \parallel K)}$	(S.4)	// by (S.3), (A.5), and jurisdiction rule
$GW \mid\equiv A_i$	(S.5)	// by (S.4) and as $A_i = (ID_i \parallel h(ID_i \parallel K) \parallel T_1)_{h(ID_i \parallel K)}$ (G.1)
$U_i \triangleleft \{Permit_{in}, T_{Permit}\}_{K_{U_i\text{-}GW}}, T_{Permit}$	(S.6)	// by (M.2)
$U_i \mid\equiv \#(U_i \overset{K_{U_i\text{-}GW}}{\longleftrightarrow} GW), U_i \mid\equiv U_i \overset{K_{U_i\text{-}GW}}{\longleftrightarrow} GW$	(S.7)	// Once (M.2) is received, U_i calculates $K_{U_i\text{-}GW}$
$U_i \mid\equiv GW \mid\sim Permit_{in}$	(S.8)	// by (S.6), (S.7), and message-meaning rule
$U_i \mid\equiv GW \mid\equiv Permit_{in}$	(S.9)	// by (S.8), (A.2), freshness-conjuncatenation rule, and nonce-verification rule
$U_i \mid\equiv Permit_{in}$	(S.10)	// by (S.9), (A.6), and jurisdiction rule (G.2)

1% to 5% of the total energy of the battery is dedicated for user authentication. It shows that a sensor node can survive from 20.2 to 100.8 months depending on the number of authentications (1 and 5 per minute) using only 1% of the total energy of the sensor node. The same figure also indicates that the sensor node can survive up to 503.9 months using 5% of the total energy.

Figure 13 also shows the number of months a sensor node can survive executing 1 and 5 online user authentications per minute using 1% to 5% of the total energy and using the MC68328 microprocessor. It shows that a sensor node can survive from 13.0 to 65 months depending on the number of authentications per minute using only 1% of the total energy of the sensor node. The same figure also indicates that the sensor node can survive up to 325 months using 5% of the total energy.

Note. If high demand of user authentication (number of authentication request per minute) in online user authentication protocol was required, the following solution could be provided. It is possible to establish a predefined session period to maintain the authentication session between the gateway node and sensor node. So, if another user requests for authentication for such node, the authentication can be

Table 23

$S_n \triangleleft (ID_i \parallel h(ID_i \parallel K_n \parallel T_{permit}) \parallel T_1)_{h(ID_i \parallel K_n \parallel T_{permit})}$	(S.1)	// by (M.1)
$S_n \mid\equiv U_i \mid\sim (ID_i \parallel h(ID_i \parallel K_n \parallel T_{permit}) \parallel T_1)_{h(ID_i \parallel K_n \parallel T_{permit})}$	(S.2)	// by (S.1), (A.4), and message-meaning rule
$S_n \mid\equiv U_i \mid\equiv (ID_i \parallel h(ID_i \parallel K_n \parallel T_{permit}) \parallel T_1)_{h(ID_i \parallel K_n \parallel T_{permit})}$	(S.3)	// by (S.2), (A.1), freshness-conjuncatenation rule, and nonce-verification rule
$S_n \mid\equiv (ID_i \parallel h(ID_i \parallel K_n \parallel T_{permit}) \parallel T_1)_{h(ID_i \parallel K_n \parallel T_{permit})}$	(S.4)	// by (S.3), (A.5), and jurisdiction rule
$S_n \mid\equiv P_i$	(S.5)	// by (S.4) and as $P_i = (ID_i \parallel h(ID_i \parallel K_n \parallel T_{permit}) \parallel T_1)_{h(ID_i \parallel K_n \parallel T_{permit})}$ (G.1)
$U_i \triangleleft (ID_i \parallel h(ID_i \parallel K_n \parallel T_{permit}) \parallel T_2)_{h(ID_i \parallel K_n \parallel T_{permit})}$	(S.6)	// by (M.2)
$U_i \mid\equiv S_n \mid\sim (ID_i \parallel h(ID_i \parallel K_n \parallel T_{permit}) \parallel T_2)_{h(ID_i \parallel K_n \parallel T_{permit})}$	(S.7)	// by (S.6), (A.3), and message-meaning rule
$U_i \mid\equiv S_n \mid\equiv (ID_i \parallel h(ID_i \parallel K_n \parallel T_{permit}) \parallel T_2)_{h(ID_i \parallel K_n \parallel T_{permit})}$	(S.8)	// by (S.7), (A.2), freshness-conjuncatenation rule, and nonce-verification rule
$U_i \mid\equiv (ID_i \parallel h(ID_i \parallel K_n \parallel T_{permit}) \parallel T_2)_{h(ID_i \parallel K_n \parallel T_{permit})}$	(S.9)	// by (S.8), (A.6), and jurisdiction rule
$U_i \mid\equiv Q_i$	(S.10)	// by (S.9) and as $Q_i = (ID_i \parallel h(ID_i \parallel K_n \parallel T_{permit}) \parallel T_1)_{h(ID_i \parallel K_n \parallel T_{permit})}$ (G.2)
$S_n \mid\equiv \#(K_{U_i\text{-}S_n}), S_n \mid\equiv K_{U_i\text{-}S_n}$	(S.11)	Once (S.5) is verified, S_n calculates $K_{U_i\text{-}S_n}$ (G.3)
$U_i \mid\equiv \#(K_{U_i\text{-}S_n}), U_i \mid\equiv K_{U_i\text{-}S_n}$	(S.12)	Once (S.10) is verified, U_i calculates $K_{U_i\text{-}S_n}$ (G.4)

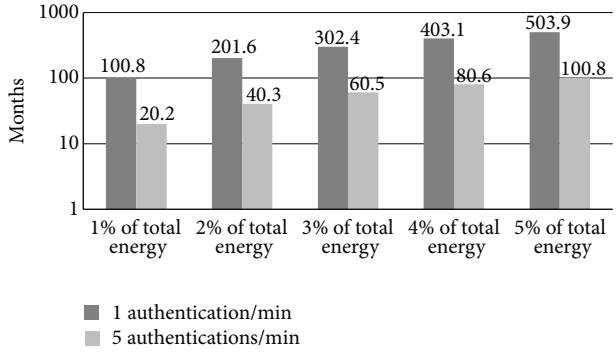


Figure 12: Number of months that a sensor node can survive executing the proposed online user authentication protocol using the MIPS R4000 microprocessor.

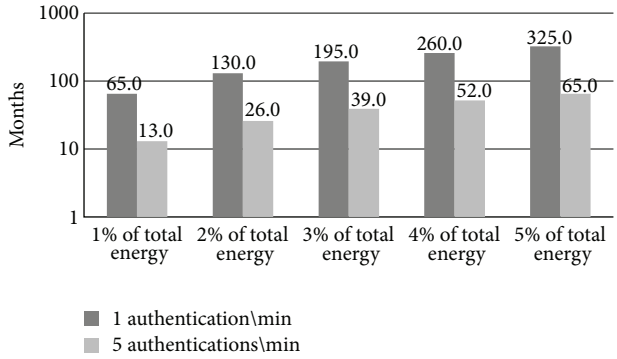


Figure 13: Number of months that a sensor node can survive executing the proposed online user authentication protocol using the MC68328 microprocessor.

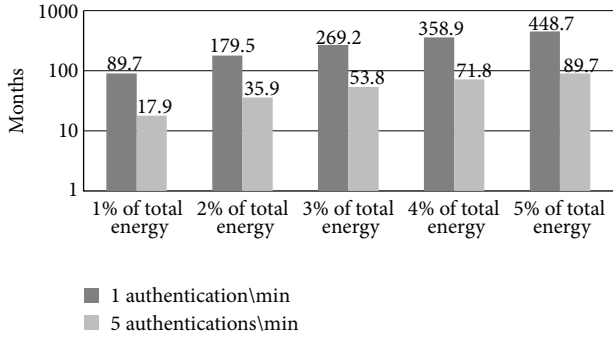


Figure 14: Number of months that a sensor node can survive executing the proposed offline user authentication sub-protocol using the MIPS R4000 microprocessor.

executed only between the U_i and GW and reuse the connection between GW and S_n created before in the previous user authentication.

Figures 14 and 15 show the number of months for which a sensor node with MIPS R4000 and MC68328 microprocessor can survive when executing 1 and 5 offline user authentications per minute assuming that 1% to 5% of the total energy of the battery is dedicated for user authentication. It shows that the survival period of a sensor node (from 11 to 448.7 months) depends on the microprocessor, number of authentication per minute, and percentage of total energy dedicated for user authentication. Once again, we believe this durability is consistent for real implementation because offline user authentication is not executed all the time but only in extraordinary situations where online user authentication cannot cover.

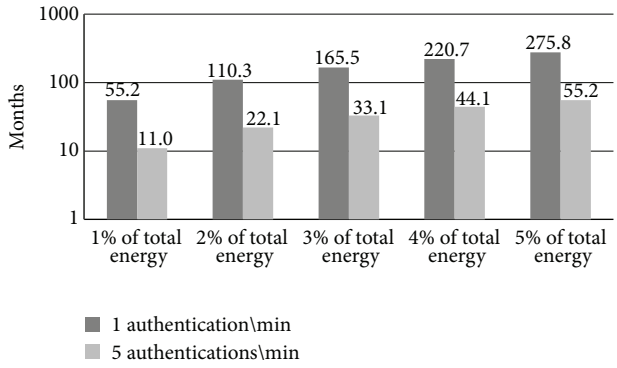


Figure 15: Number of months that a sensor node can survive executing the proposed offline user authentication sub-protocol using the MC68328 microprocessor.

6. Conclusion

With the increase of different types of sensor network implementations such as medical, ecology, and military operation applications, there have been many proposals which tried to give secure user authentication schemes for them. However, even though they deliver important advance in this area, they still incorporate serious vulnerabilities and limitations. In those circumstances, this paper proposes a user authentication mechanism which considers the security, performance, and usability factors. The security is guaranteed by an intensive analysis in terms of formal verification and analysis of possible attacks. The optimization of performance is achieved by using lightweight cryptography and in most of cases, only hash functions and XOR operations; additionally, the number of messages is reduced by using timestamps instead of challenge response of random nonces. Finally, the usability requirements are satisfied by considering both the online and offline user authentication use cases, the feature which was not considered in previous works.

In summary, this paper analyzes previous user authentication mechanisms for wireless sensor networks and identifies their vulnerabilities and limitations and proposes a robust user authentication for wireless sensor networks that eliminates the identified security flaws and limitations. The proposed solution takes advantage of the two-factor authentication concept to provide a secure authentication system offering balanced features in terms of security, performance, and usability.

Acknowledgment

This work was supported by the Sogang University Research Grant of 2011 (no. 201110026).

References

[1] M. H. Eldefrawy, M. K. Khan, K. Alghathbar, and E. S. Cho, "Broadcast authentication for wireless sensor networks using nested hashing and the chinese remainder theorem," *Sensors*, vol. 10, no. 9, pp. 8683–8695, 2010.

[2] A. Hagedorn, D. Starobinski, and A. Trachtenberg, "Rateless Deluge: over-the-air programming of wireless sensor networks using random linear codes," in *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN '08)*, pp. 457–466, April 2008.

[3] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.

[4] D. Nyang and M. Lee, "Improvement of Das's two-factor authentication protocol in wireless sensor networks," Cryptology ePrint archive 2009/631, 2012, http://eprint.iacr.org/2009/631.pdf.

[5] H. F. Huang, Y. F. Chang, and C. H. Liu, "Enhancement of two-factor user authentication in wireless sensor networks," in *Proceedings of the 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP '10)*, pp. 27–30, October 2010.

[6] T. H. Chen and W. K. Shih, "A robust mutual authentication protocol for wireless sensor networks," *ETRI Journal*, vol. 32, no. 5, pp. 704–712, 2010.

[7] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks," *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.

[8] S. Yoo, K. Y. Park, and J. Kim, "A security-performance balanced user authentication scheme for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 382810, 11 pages, 2012.

[9] G. Werner-Allen, K. Lorincz, M. Welsh et al., "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, 2006.

[10] D. Doley and A. C. Yao, "On the security of public-key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[11] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proceedings of the 19th International Advances in Cryptography Conference (CRYPTO '99)*, pp. 388–397, 1999.

[12] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[13] B. Jack, "Exploiting embedded systems," Black Hat, 2006, http://www.blackhat.com/presentations/bh-europe-06/bh-eu-06-Jack.pdf.

[14] M. E. Raluca, M. E. Razvan, and A. Terzis, "Gateway design for data gathering sensor networks," in *Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '08)*, pp. 296–304, June 2008.

[15] National Instruments, WSN Ethernet Gateway, 2012, http://sine.ni.com/nips/cds/view/p/lang/en/nid/206919.

[16] D. W. Carman, P. S. Krus, and B. J. Matt, "Constraints and approaches for distributed sensor network security," Tech. Rep. 00-010, NAI Labs, Network Associates, Glenwood, Md, USA, 2000.

[17] G. Bertoni, L. Breveglieri, and M. Venturi, "Power aware design of an elliptic curve coprocessor for 8 bit platforms," in *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops '06)*, pp. 337–341, March 2006.

[18] D. J. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic

curve cryptography," in *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON '04)*, pp. 71–80, October 2004.

[19] K. Piotrowski, P. Langendoerfer, and S. Peter, "How public key cryptography influences wireless sensor node lifetime," in *Proceedings of the 4th ACM Workshop on Security of ad hoc and Sensor Networks (SASN '06)*, pp. 169–176, October 2006.

[20] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proceedings of the Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops '05)*, pp. 324–328, 2005.

[21] C. C. Chang, D. J. Nagel, and S. Muftic, "Balancing security and energy consumption in wireless sensor networks," *Lecture Notes in Computer Science*, vol. 4864, pp. 469–480, 2007.

[22] J. P. Kaps and B. Sunar, "Energy comparison of AES and SHA-1 for ubiquitous computing," *Lecture Notes in Computer Science*, vol. 4097, pp. 372–381, 2006.

[23] M. Burrows, M. Abadi, and R. Needham, "Logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[24] Y. G. Kim, C. J. Moon, D. W. Jeong, and D. K. Baik, "Formal verification of bundle authentication mechanism in OSGi service platform: BAN Logic," *International Journal of Software Engineering and Knowledge Engineering*, vol. 16, no. 2, pp. 153–173, 2006.

[25] W. Tsaur, J. Li, and W. Lee, "An efficient and secure multi-server authentication scheme with key agreement," *Journal of Systems and Software*, vol. 85, no. 4, pp. 876–882, 2012.

[26] S. Wang, Q. Ma, Y. Zhang, and Y. Li, "An Authentication protocol for RFID tag and its simulation," *Journal of Networks*, vol. 6, no. 3, pp. 446–453, 2011.

[27] J. L. Tsai, T. C. Wu, and K. Y. Tsai, "New dynamic ID authentication scheme using smart cards," *International Journal of Communication Systems*, vol. 23, no. 12, pp. 1449–1462, 2010.

[28] C. Gamage, J. Leiwo, and Y. Zheng, *Timestamps for Network Authentication Protocols Revisited*, 1999.

[29] R. Ying, "Building systems using software components," *Journal of Software Technology*, vol. 9, no. 1, 2006.

[30] J. Park and S. Sahni, "An online heuristic for maximum lifetime routing in wireless sensor networks," *IEEE Transactions on Computers*, vol. 55, no. 8, pp. 1048–1056, 2006.

[31] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[32] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[33] S. G. Yoo, S. H. Kang, and J. Kim, "SERA: a secure energy reliability aware data gathering for sensor networks," *Multimedia Tools and Applications*, pp. 1–30, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 375953, 6 pages
http://dx.doi.org/10.1155/2013/375953

Research Article

A Novel Verifiable Multisecret Sharing Scheme in Wireless Multimedia Sensor Networks

Xianfeng Guo[1] and Qing Luo[2]

[1] *College of Computer Science and Technology, Southwest University for Nationalities, Chengdu 610041, China*
[2] *Sichuan Conservatory of Music, Chengdu 610021, China*

Correspondence should be addressed to Xianfeng Guo; guoxianf@126.com

Received 25 January 2013; Accepted 9 April 2013

Academic Editor: Muhammad Khurram Khan

This paper presents a secure and practical verifiable multi-secret sharing scheme with application on multimedia sensor networks. By utilizing the merits of biometric cryptography, the proposed scheme can reliably conceal private information and overcome the shortcoming of the traditional cryptography. Security analysis shows that the proposed scheme provides secure, robust, and trustworthy verification of dealer and participants over insecure wireless multimedia sensor networks.

1. Introduction

Wireless multimedia sensor networks (WMSNs) have started to receive a lot of attention very recently due to their potential to be deployed flexibly in various applications with lower costs [1, 2]. As WMSNs are widely deployed in remote and hostile environments to transmit sensitive information by broadcast, sensor nodes are prone to node compromise attacks and security issues such as data confidentiality and integrity are extremely important. Hence, security becomes a very serious concern in WMSNs protocols. But unfortunately sensor nodes have limited power, computation, storage, and communication capabilities; they impose several constraints on the algorithm and protocol that can be effectively deployed for such systems. In this scenario, most of the traditional security mechanisms are useless. Thus, the research of new efficient security techniques is needed [3–5].

In order to keep the secret efficiently and safely, in 1979, Shamir [6] and Blakley [7] first developed the concepts of the secret sharing (SS) scheme, respectively. In such a scheme, the dealer splits the secret into shares among participants, and sends the share to the corresponding participant. As a result, any t out of the n participants can cooperate to resume the secret, but any less than t out of the n participants cannot get any useful information about the secret by any way. A threshold secret sharing scheme has many practical applications, such as opening a bank vault, launching a nuclear, or authenticating an electronic funds transfer. Later, several multisecret sharing schemes were proposed [8–12]. In a multisecret sharing scheme, there are multiple secrets to be shared during one secret sharing process. In 2004, Yang et al. (YCH) [13] proposed a novel multisecret sharing (MSS) scheme, which is based on a two-variable one-way function. Besides efficient computation for practice, the scheme has the following merits: (1) several secrets can be shared during one secret sharing process; (2) the dealer need not redistributes a fresh shadow after the secrets have been reconstructed. But YCH scheme does not have the property of verification. That is to say, neither dealer nor participants cheating can be detected. As for dealer cheating, the dishonest dealer can prevent any certain participant from obtaining the true secret by distributing a fake shadow to him/her [14]; and the participants cheating, a malicious participant may provide a fake shadow to other participants, which makes the malicious participant the only one who gets to reconstruct the true secret [15]. In a secret sharing involving multiple dealers, the property of verifiability is more desirable since these dealers are mutually distrusted.

In 2005, Shao and Cao (SC) [16] proposed a verifiable multisecret sharing (VMSS) based on YCH and the intractability of the discrete logarithm. However, the speed of SC scheme is not competitive [17]. In 2007, Zhao et al.

(ZZZ) [18] proposed another practical VMSS based on YCH. ZZZ scheme perform the verification phase by utilizing RSA cryptosystem and a Diffie-Helman key agreement method. Though ZZZ scheme significantly reduced the computation costs of the whole system, it is required that the dealer reconstructs a RSA cryptosystem once the secrets have been reconstructed. Later on, Dehkordi and Mashhadi (DM) [17] proposed an efficient VMSS scheme to improve the performance of SC scheme. Compared to ZZZ scheme, DM scheme just constructs one RSA cryptosystem in the whole system and the dealer verification is not required. But any participant has to compute $t-1$ modular exponential computing to verify others' secret shadow. It is still very time-consuming. In addition, it is difficult to protect private keys from attackers in the password-based verification system [19].

To overcome the drawback and pitfall of only-password-based verification of YCH scheme, in this paper, we present an efficient and practical VMSS by using biometric data, for example, palmprint, with application on wireless multimedia sensor networks. In the verification phase of the proposed scheme, the time consuming modular exponentiation computations, for example, Diffie-Hellman or RSA, are eliminated. This scheme allows participants and dealer to cooperate in choosing secret shadows. It can effectively perform dealer and participants' verification. Moreover, the computation cost, security, and efficiency of the presented scheme are embarking for the real application in the practical environment.

Rest of the paper is organized as follows: Section 2 briefly reviews YCH scheme. Section 3 presents the efficient and practical palmprint-based VMSS scheme with application on wireless multimedia sensor networks. Section 4 performs performance analysis of the proposed scheme. Conclusions are given in Section 5.

2. Review of YCH Schemes

Function $f(r,x)$ denotes any two-variable one-way Hash function and $(P_1, P_2, \ldots, P_k)$ denotes k secrets to be shared among n participants. Before the secret sharing, the dealer randomly chooses n secret shadows $s_1, s_2, \ldots, s_n$ and distributes them to every participant over a secret channel. Then the dealer performs the following steps.

(2.1) *If* $k \le t$.

(a) Chooses a big prime Q and constructs $(t-1)$th degree polynomial $h(x) \bmod Q$:

$$h(x) = P_1 + P_2 x + \cdots + P_k x^{k-1} + a_1 x^k + a_2 x^{k+1} + \cdots + a_{t-k} x^{t-1} \bmod Q, \tag{1}$$

where $0 < P_1, P_2, \ldots, P_k, a_1, a_2, \ldots, a_{t-k} < Q$.

(b) Compute $y_i = h(f(r, s_i)) \bmod Q$ for $i = 1, 2, \ldots, n$.

(c) Publish $(r, y_1, y_2, \ldots, y_n)$.

(2.2) *If* $k > t$.

(a) Choose a big prime Q and construct $(k-1)$th degree polynomial $h(x) \bmod Q$:

$$h(x) = P_1 + P_2 x + \cdots + P_k x^{k-1} \bmod Q, \tag{2}$$

where $0 < P_1, P_2, \ldots, P_k < Q$.

(b) Compute $y_i = h(f(r, s_i)) \bmod Q$ for $i = 1, 2, \ldots, n$.

(c) Compute $h(i) \bmod Q$ for $i = 1, 2, \ldots, k-t$.

(d) Publish $(r, y_1, y_2, \ldots, y_n, h(1), h(2), \ldots, h(k-t))$.

If at least t participants pool their pseudo shadows $f(r, s_i)$ (for $i = 1, 2, \ldots, t$), then the polynomial $h(x) \bmod Q$ can be uniquely determined as follows.

(1) *If* $k \le t$

$$\begin{aligned} h(x) &= \sum_{i=1}^{t} y_i \prod_{j=1, j\neq i}^{t} \frac{x - f(r, s_j)}{f(r, s_i) - f(r, s_j)} \bmod Q \\ &= P_1 + P_2 x + \cdots + P_k x^{k-1} + a_1 x^k \\ &\quad + a_2 x^{k+1} + \cdots + a_{t-k} x^{t-1} \bmod Q. \end{aligned} \tag{3}$$

(2) *If* $k > t$

$$\begin{aligned} h(x) &= \sum_{i=1}^{t} y_i \prod_{j=1, j\neq i}^{t} \frac{x - f(r, s_j)}{f(r, s_i) - f(r, s_j)} \\ &\quad \times \prod_{l=1, l\neq f(r,s_i)}^{k-t} \frac{x - l}{f(r, s_i) - l} \\ &\quad + \sum_{i=1}^{k-t} h(i) \prod_{j=1, j\neq i}^{k-t} \frac{x - j}{i - j} \\ &\quad \times \prod_{l=1, l\neq f(r,s_i)}^{t} \frac{x - f(r, s_l)}{i - f(r, s_l)} \bmod Q \\ &= P_1 + P_2 x + \cdots + P_k x^{k-1} \bmod Q. \end{aligned} \tag{4}$$

Then the shared secrets $P_1, P_2, \ldots, P_k$ are recovered.

3. The Proposed Palmprint-Based VMSS Scheme

In this section we will propose a new palmprint-based VMSS scheme with application on wireless multimedia sensor networks. The verification is based on matching of the palmprint template. The notations $(P_1, P_2, \ldots, P_k)$ and $f(r,x)$ in this scheme are the same as those of Section 2. Define D and $M = \{M_i\}_{i=1}^{n}$ are dealer (Server) and set of participants respectively.

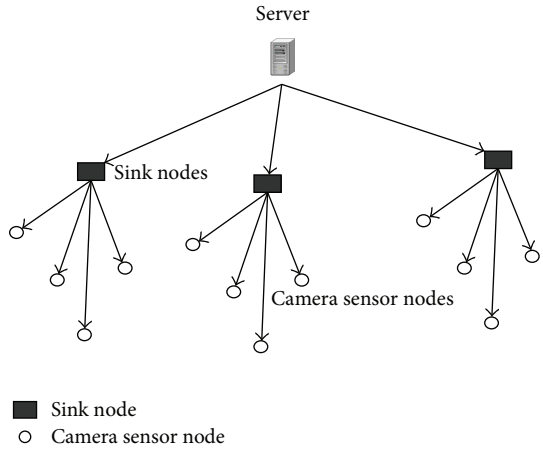


Figure 1: System architecture of the multimedia sensor network with VMSS.

3.1. Architecture of Wireless Multimedia Sensor Networks. Figure 1 shows the developed architecture of the secure wireless multimedia sensor networks by utilizing the proposed biometric based VMSS. Each camera sensor node in the networks is battery-powered and has limited computation and wireless communication capabilities. The sink is a data collection and storage center equipped with sufficient computation and storage capabilities. Camera sensor nodes periodically send the captured images to the sink node. Then the sink nodes transport this information secretly with the data process server via carrier networks. The proposed scheme is mounting among the sink nodes.

3.2. Initialization Phase. Figure 2 shows an overall flowchart of the initialization phase. For $i = 1, 2, \ldots, n$ participant M_i first randomly chooses his private key s_i' and submits to the dealer D. M_i also imprints his palmprint image at the camera sensor nodes, and then initialization system performs the following operations.

(1) Dealer D randomly chooses a corresponding secret shadow s_i and computes $f(s_i', s_i)$.

(2) Write M_i, B_i and $V_i = f(s_i', s_i) \oplus B_i$ into the memory of the corresponding sink node, where B_i is the palmprint template of M_i. For more details on calculate $V_i = f(s_i', s_i) \oplus B_i$, the reader can refer to [20].

3.3. Construction Phase. The dealer D chooses a random number r and performs the following steps.

(1) *If* $k \le t$.

(a) Chooses a big prime Q and constructs $(t-1)$th degree polynomial $h(x)$ mod Q:

$$h(x) = P_1 + P_2 x + \cdots + P_k x^{k-1} + a_1 x^k + a_2 x^{k+1} + \cdots + a_{t-k} x^{t-1} \bmod Q, \quad (5)$$

where $0 < P_1, P_2, \ldots, P_k, a_1, a_2, \ldots, a_{t-k} < Q$.

(b) Compute $y_i = h(f(r, f(s_i', s_i))) \bmod Q$ for $i = 1, 2, \ldots, n$.

(c) Publish $(r, y_1, y_2, \ldots, y_n)$.

(2) *If* $k > t$.

(a) Choose a big prime Q and construct $(k-1)$th degree polynomial $h(x)$ mod Q:

$$h(x) = P_1 + P_2 x + \cdots + P_k x^{k-1} \bmod Q, \quad (6)$$

where $0 < P_1, P_2, \ldots, P_k < Q$.

(b) Compute $y_i = h(f(r, f(s_i', s_i))) \bmod Q$ for $i = 1, 2, \ldots, n$.

(c) Compute $h(i) \bmod Q$ for $i = 1, 2, \ldots, k-t$.

(d) Publish $(r, y_1, y_2, \ldots, y_n, h(1), h(2), \ldots, h(k-t))$.

3.4. Verification Phase. By utilizing the published r and the secret shadow s_i, each participant $\{M_i\}_{i=1}^n$ can compute his secret share $f(r, f'(s_i', s_i))$. If participants $\{M_i\}_{i=1}^t$ pool their shares $f(r, f'(s_i', s_i))$ and identity M_i, dealer D can perform the following verifications.

Computes $f(r, f(s_i', s_i))$ and compares $f(r, f(s_i', s_i)) \stackrel{?}{=} f(r, f'(s_i', s_i))$. If they are equal, D believes that M_i is honest.

Furthermore, $f(s_i', s_i)$ is composed of the private key s_i' of M_i and the secret shadow s_i of D, therefore the dealer verification is unnecessary.

3.5. Recovery Phase. The security of a recovery phase is based on the palmprint cryptosystem. Suppose participants $\{M_i\}_{i=1}^t$ want recovery the shared secret $(P_1, P_2, \ldots, P_k)$, M_i (for $i = 1, 2, \ldots, t$) imprints his palmprint to the camera sensor and the recovery system performs the following operations (the framework of this phase are shown in Figure 3).

(1) By utilizing the input palmprint image, generate M_i's palmprint template B_i'.

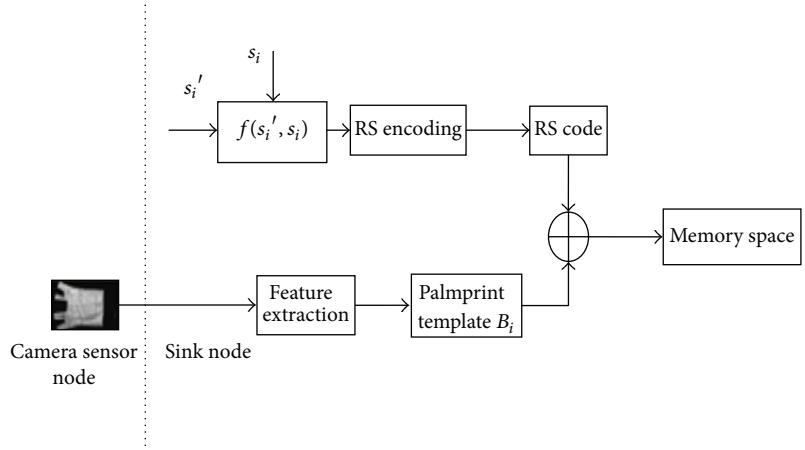


Figure 2: Overall flowchart of the initialization phase.

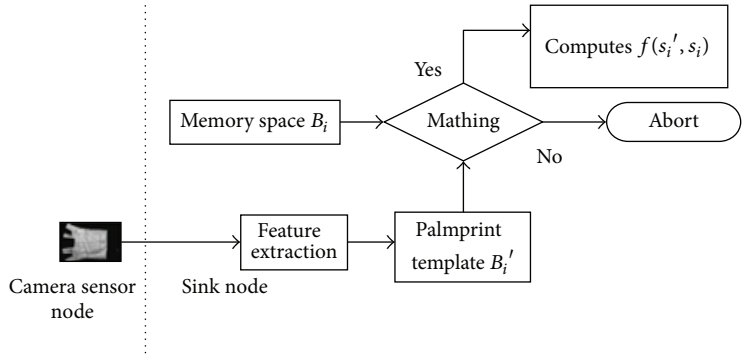


Figure 3: Framework of the recovery phase.

(2) The sink node then performs matching process of B_i' with B_i. If yes, compute $f(s_i', s_i) = V_i \oplus B_i$ and send $(f(r, f(s_i', s_i)), M_i)$ to the recovery system. If no, the recovery is aborted.

By utilizing the secret shadows $f(r, f(s_i', s_i))$ (for $i = 1, 2, \ldots, t$) and the corresponding published values, the polynomial $h(x)$ mod Q can be uniquely determined as follows.

(1) *If* $k \le t$

$$\begin{aligned} h(x) &= \sum_{i=1}^{t} y_i \prod_{j=1, j\neq i}^{t} \frac{x - f(r, s_j)}{f(r, s_i) - f(r, s_j)} \bmod Q \\ &= P_1 + P_2 x + \cdots + P_k x^{k-1} + a_1 x^k \\ &\quad + a_2 x^{k+1} + \cdots + a_{t-k} x^{t-1} \bmod Q. \end{aligned} \tag{7}$$

(2) *If* $k > t$

$$\begin{aligned} h(x) &= \sum_{i=1}^{t} y_i \prod_{j=1, j\neq i}^{t} \frac{x - f(r, s_j)}{f(r, s_i) - f(r, s_j)} \\ &\quad \times \prod_{l=1, l\neq f(r, s_i)}^{k-t} \frac{x - l}{f(r, s_i) - l} \\ &\quad + \sum_{i=1}^{k-t} h(i) \prod_{j=1, j\neq i}^{k-t} \frac{x - j}{i - j} \\ &\quad \times \prod_{l=1, l\neq f(r, s_i)}^{t} \frac{x - f(r, s_l)}{i - f(r, s_l)} \bmod Q \\ &= P_1 + P_2 x + \cdots + P_k x^{k-1} \bmod Q. \end{aligned} \tag{8}$$

Then the shared secrets $P_1, P_2, \ldots, P_k$ are recovered.

4. Performance Analysis

4.1. Security Analysis. The security of our proposed scheme is based on (t, n) Shamir scheme and palmprint cryptography. In the rest of this section, some possible attacks will be raised and fought against to demonstrate the security of our scheme.

(1) If attacker Eve can use fewer points than t to reconstruct the polynomial $h(x)$ mod Q, it is equal to Eve that has broken through the (t, n) Shamir's scheme successfully. In this point, it is infeasible to recovery the secret shares of the proposed scheme by fewer points than t.

Table 1: Computation quality of ZZZ, DM and our scheme.

By	For	ZZZ scheme	DM scheme	Our scheme
M_i	Initialization	$R_i = g^{si} \bmod N$	$s_i^e \bmod N$	—
Dealer	Construction	Construct a RSA cryptosystem	One Hash function	One Hash function
M_i	Recovery	$I_i = R_0^{si} \bmod N$	One Hash function	One Hash function
M_i	Verification	$\{I_j^f \stackrel{?}{=} R_j \bmod N\}_{j=1,j\neq i}^{t}$	t exponentiations	t Hash function

(2) In this scheme, the dealer D cheating would be failed, because every secret shadow $f(s_i', s_i)$ is composed of the private key s_i' of participant $\{M_i\}_{i=1}^n$ and the secret shadow s_i of D, either subtle differences of the secret will change $f(s_i', s_i)$ significantly. This is based on the sensitivity to initial condition of Hash function $f(r, x)$.

(3) Suppose dishonest participant M_i gives a fake share $f'(r, f(s_i', s_i))$ to prevent the shared secrets recovery, it will be detected by dealer D in verification phase. Dealer D can compare $f(r, f(s_i', s_i)) \stackrel{?}{=} f(r, f'(s_i', s_i))$ to insure its correctness. The security is depending on the collision-free property of chaotic Hash function $f(r, x)$.

(4) Replay of the former round secret share $f(r, f(s_i', s_i))$ can be exposed, because r is random selected in each round and Hash function $f(r, x)$ is sensitivity to initial condition.

(5) Any intruder cannot get $f(s_i', s_i)$ because they cannot match the palmprint from the sensor nodes.

(6) The same as (5), the intruder cannot compute the secret share $f(r, f(s_i', s_i))$ from the sink's data.

4.2. Computation Quality. Compared with the proposed YCH based VMSS schemes, such as DM scheme [17], ZZZ scheme [18] and SC scheme [16], the verification phase of our scheme is more efficient and suitable for using in the WMSNs. SC scheme requires some time consuming power products in the verification phase, so it is too complex to applicable on WMSNs. ZZZ scheme performs the verification phase by utilizing RSA cryptosystem and a Diffie-Helman key agreement method. It still requires $t + 1$ exponentiations to verify the validity of the secret shadows. And DM scheme just needs $t - 1$ modular exponential computing to verify others' secret shadow. The verification of our scheme is based on the collision-free one-way Hash function. Any participant requires only one chaotic Hash computation to verify his/her shadow's validity, and $t - 1$ Hash computations to verify other participants' shadows. Therefore, our scheme is more convenient for application on the WMSNs devices. Table 1 is for the comparison of these schemes.

5. Conclusion

We have proposed a novel and complete biometric-based verifiable multisecret sharing scheme with application on WMSNs. This scheme can conceal the secret shadow by palmprint cryptography and provide perfect verification. Performance analyses demonstrate it can achieve more functionality and satisfies all criteria.

Acknowledgments

This work described here was supported by the Fundamental Research Funds for the Central Universities, Southwest University for Nationalities (nos. 11NZYQN27 and 11NZYTH06), and the National Natural Science Foundation of China (no. 61105061).

References

[1] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of "two-factor user authentication in wireless sensor networks"," *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.

[2] X. Guo, J. Zhang, M. K. Khan, and K. Alghathbar, "Secure chaotic map based block cryptosystem with application to camera sensor networks," *Sensors*, vol. 11, no. 2, pp. 1607–1619, 2011.

[3] X. He, M. Niedermeier, and H. de Meer, "Dynamic key management in wireless sensor networks: a survey," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 611–622, 2013.

[4] Q. Shi, N. Zhang, M. Merabti, and K. Kifayat, "Resource-efficient authentic key establishment in heterogeneous wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 73, no. 2, pp. 235–249, 2013.

[5] S. Qian, "A novel key pre-distribution for wireless sensor networks," *Physics Procedia*, vol. 25, pp. 2183–2189, 2012.

[6] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[7] G. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the AFIPS National Computer Conference*, pp. 313–317, AFIPS Press, 1979.

[8] C.-W. Chan and C.-C. Chang, "A scheme for threshold multi-secret sharing," *Applied Mathematics and Computation*, vol. 166, no. 1, pp. 1–14, 2005.

[9] Y.-F. Chen, Y.-K. Chan, C.-C. Huang, M.-H. Tsai, and Y.-P. Chu, "A multiple-level visual secret-sharing scheme without image size expansion," *Information Sciences*, vol. 177, no. 21, pp. 4696–4710, 2007.

[10] H.-Y. Chien, J.-K. Jan, and Y.-M. Tseng, "A practical (*t, n*) multi-secret sharing scheme," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 83, no. 12, pp. 2762–2765, 2000.

[11] J. He and E. Dawson, "Multisecret-sharing scheme based on one-way function," *Electronics Letters*, vol. 31, no. 2, pp. 93–95, 1995.

[12] M. Liu, L. Xiao, and Z. Zhang, "Linear multi-secret sharing schemes based on multi-party computation," *Finite Fields and their Applications*, vol. 12, no. 4, pp. 704–713, 2006.

[13] C.-C. Yang, T.-Y. Chang, and M.-S. Hwang, "A (*t,n*) multi-secret sharing scheme," *Applied Mathematics and Computation*, vol. 151, no. 2, pp. 483–490, 2004.

[14] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch, "Verifiable secret sharing and achieving simultaneity in the presence of faults," in *Proceedings of the 26th IEEE Symposium on the Foundations of Computer Science (FOCS '85)*, pp. 251–260, 1985.

[15] M. Tompa and H. Woll, "How to share a secret with cheaters," *Journal of Cryptology*, vol. 1, no. 3, pp. 133–138, 1989.

[16] J. Shao and Z.-F. Cao, "A new efficient (*t, n*) verifiable multi-secret sharing (VMSS) based on YCH scheme," *Applied Mathematics and Computation*, vol. 168, no. 1, pp. 135–140, 2005.

[17] M. H. Dehkordi and S. Mashhadi, "An efficient threshold verifiable multi-secret sharing," *Computer Standards and Interfaces*, vol. 30, no. 3, pp. 187–190, 2008.

[18] J. Zhao, J. Zhang, and R. Zhao, "A practical verifiable multi-secret sharing scheme," *Computer Standards and Interfaces*, vol. 29, no. 1, pp. 138–141, 2007.

[19] M. K. Khan, J. Zhang, and L. Tian, "Chaotic secure content-based hidden transmission of biometric templates," *Chaos, Solitons and Fractals*, vol. 32, no. 5, pp. 1749–1759, 2007.

[20] X. Wu, K. Wang, and D. Zhang, "A cryptosystem based on palmprint feature," in *Proceedings of the 19th International Conference on Pattern Recognition (ICPR '08)*, December 2008.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 950704, 12 pages
http://dx.doi.org/10.1155/2013/950704

Review Article

A Review of IEEE 802.15.6 MAC, PHY, and Security Specifications

Sana Ullah,[1,2] Manar Mohaisen,[3] and Mohammed A. Alnuem[1,2]

[1] *Department of Information System, College of Computer and Information Science, King Saud University, Riyadh 11543, Saudi Arabia*
[2] *Chair of Pervasive and Mobile Computing, King Saud University, Riyadh 11543, Saudi Arabia*
[3] *Department of EEC Engineering, Korea Tech, Cheonan 330-708, Republic of Korea*

Correspondence should be addressed to Sana Ullah; sanajcs@gmail.com

Received 24 December 2012; Accepted 12 March 2013

Academic Editor: Muhammad Khurram Khan

IEEE 802.15.6 is the first international Wireless Body Area Network (WBAN) standard that supports communications in the vicinity of or inside a human body to serve a variety of medical and nonmedical applications. The standard defines a Medium Access Control (MAC) layer that supports several Physical (PHY) layers. In this paper, the key features of the IEEE 802.15.6 standard are presented. The MAC, PHY, and security specifications of the standard are explained in detail. Different communication modes and access mechanisms are presented. The Narrowband (NB), Ultra-wideband (UWB), and Human Body Communications (HBC) specifications are reviewed in terms of frame structure, modulation, and other important parameters. Finally, the security paradigm and services of the standard are highlighted.

1. Introduction

Wireless Body Area Networks (WBANs) are replacing conventional healthcare systems by enabling the ambulatory and continuous health monitoring of patients suffering from chronic diseases, such as heart disease [1–3]. The recent popularity of WBANs has attracted a number of researchers from academia and industry in the past few years. The heterogeneous WBAN requirements have allowed the realisation of a standard model that can support all of the relevant requirements. In November 2007, IEEE 802 established a task group for the standardisation of WBAN called IEEE 802.15.6 [4]. Earlier, IEEE 802 had several successful experiences in developing international standards for wireless communications [5–8], including the IEEE 802.11 [9], IEEE 802.15.1 [10], and IEEE 802.15.4 [11] standards. The aim of the IEEE 802.15.6 standard was to develop a communications standard for miniaturised low-power devices that are deployed on or implanted inside a human body to serve a variety of medical, consumer electronics (CE), and entertainment applications [12]. The standardisation body received a total of 34 proposals, which were later merged into a single proposal. In July 2011, the initial draft of the standard was approved to start a sponsor ballot. The final version of the standard was published in February 2012.

The IEEE 802.15.6 standard defines a Medium Access Control (MAC) layer that supports several Physical (PHY) layers, such as Narrowband (NB), Ultra-wideband (UWB), and Human Body Communications (HBC) layers, as illustrated in Figure 1. The proper selection of PHYs or frequency bands has remained one of the important issues to be considered in the development of WBANs [13]. Generally, the available frequencies for WBANs are regulated by communication authorities in different countries. Figure 2 shows the available frequency bands for WBANs [14]. The Medical Implant Communications Service (MICS) band is a licensed band used for implant communications and has the same frequency range of 402–405 MHz in most countries. Wireless Medical Telemetry Services (WMTSs) is a licensed band used for medical telemetry systems. The problems with the MICS and WMTS bands are their inability to support high-data-rate applications. The Industrial, Scientific, and Medical (ISM) and Ultra-wideband (UWB) bands support high-data-rate applications and are available worldwide. However, there is a high probability of interference because many wireless

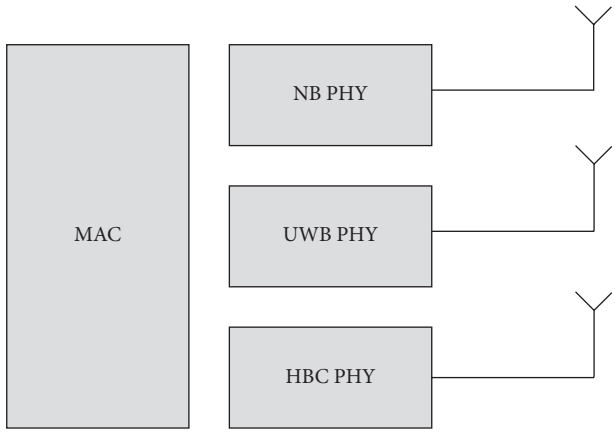


Figure 1: IEEE 802.15.6 MAC and PHY layers.

devices, including those using the IEEE 802.11 and IEEE 802.15.4 standards, operate in the 2.4 GHz band [15]. For efficient resource allocation on the channel, the standard allows the nodes and hubs (which are also known as coordinators) to establish a time reference base, where the time axis is divided into multiple superframes of equal length [16–18]. In this paper, we present the key features of the IEEE 802.15.6 standard. Starting from the fundamental details, we provide deep insight into the MAC and PHY layers and the security specifications of the standard. We review different communication modes and access mechanisms and explain the NB, UWB, and HBC specifications in detail. We also review the security paradigm and services of IEEE 802.15.6.

The remainder of this paper is organised into four sections. Sections 2 and 3 present the IEEE MAC and PHY specifications, respectively. Section 4 presents the security specifications, and Section 5 concludes our work.

2. IEEE 802.15.6 MAC Specifications

According to the IEEE 802.15.6 standard, the nodes are organised into one- or two-hop star WBANs. A single coordinator or hub controls the entire operation of each WBAN. The WBAN must have one hub and a number of nodes, ranging from zero to *mMaxBANSize*. In a two-hop start WBAN, a relay-capable node may be used to exchange data frames between a node and the hub. The standard divides the time axis or channel into beacon periods or superframes of equal length. Each superframe contains a number of allocation slots that are used for data transmission. These slots have equal duration and are numbered from 0 to s, where $s \leq 255$. The hub transmits beacons to define the superframe boundaries and allocate the slots. For nonbeacon modes, the superframe boundaries where beacons are not used are defined by polling frames. Generally, the hub transmits beacons in each superframe except those that are inactive. The hub may shift or rotate the offsets of the beacon periods, thus shifting the schedule allocation slots. The following sections present the MAC frame format, communication modes, and access mechanisms defined in the IEEE 802.15.6 standard.

2.1. IEEE 802.15.6 MAC Frame Format. Figure 3 shows the general MAC frame format consisting of a 56-bit header, variable length frame body, and 18-bit Frame Check Sequence (FCS). The maximum length of the frame body is 255 octets. The MAC header further consists of 32-bit frame control, 8-bit recipient Identification (ID), 8-bit sender ID, and 8-bit WBAN ID fields. The frame control field carries control information including the type of frame, that is, beacon, acknowledgement, or other control frames. The recipient and sender ID fields contain the address information of the recipient and the sender of the data frame, respectively. The WBAN ID contains information on the WBAN in which the transmission is active. The first 8-bit field in the MAC frame body carries message freshness information required for nonce construction and replay detection. The frame payload field carries data frames, and the last 32-bit Message Integrity Code (MIC) carries information about the authenticity and integrity of the frame.

2.2. IEEE 802.15.6 Communication Modes. The IEEE 802.15.6 supports the following communication modes.

2.2.1. Beacon Mode with Superframe Boundaries. In this mode, the hub transmits beacons in active superframes. The active superframes may be followed by several inactive superframes whenever there is no scheduled transmission. As illustrated in Figure 4(a), the superframe structure is divided into Exclusive Access Phases (EAP1 and EAP2), Random Access Phases (RAP1 and RAP2), a Managed Access Phase (MAP), and a Contention Access Phase (CAP). The EAPs are used to transfer high-priority or emergency traffic. The RAPs and CAP are used for nonrecurring traffic. The MAP period is used for scheduled and unscheduled bilink allocations, scheduled uplink and downlink allocations, and Type I (not Type II) polled and posted allocations. The length of Type I and Type II allocations is represented in terms of the transmission time and number of frames, respectively. A detailed discussion of these allocations is presented in Section 2.3.

2.2.2. Nonbeacon Mode with Superframe Boundaries. In this mode, the hub operates during the MAP period only, as illustrated in Figure 4(b).

2.2.3. Nonbeacon Mode without Superframe Boundaries. In this mode, the hub provides unscheduled Type II polled or posted allocations or a combination of both, as depicted in Figure 4(c).

2.3. IEEE 802.15.6 Access Mechanisms. The IEEE 802.15.6 supports the following access mechanisms.

2.3.1. Random Access Mechanism. In EAP, RAP, and CAP periods, the hub may employ either a slotted ALOHA or Carrier Sensor Multiple Access/Collision Avoidance (CSMA/CA) protocol, depending on the PHY. The hub considers slotted ALOHA and CSMA/CA protocols for UWB and NB PHYs, respectively. To send high-priority data frames using CSMA/CA, the hub may combine EAP1 and RAP1

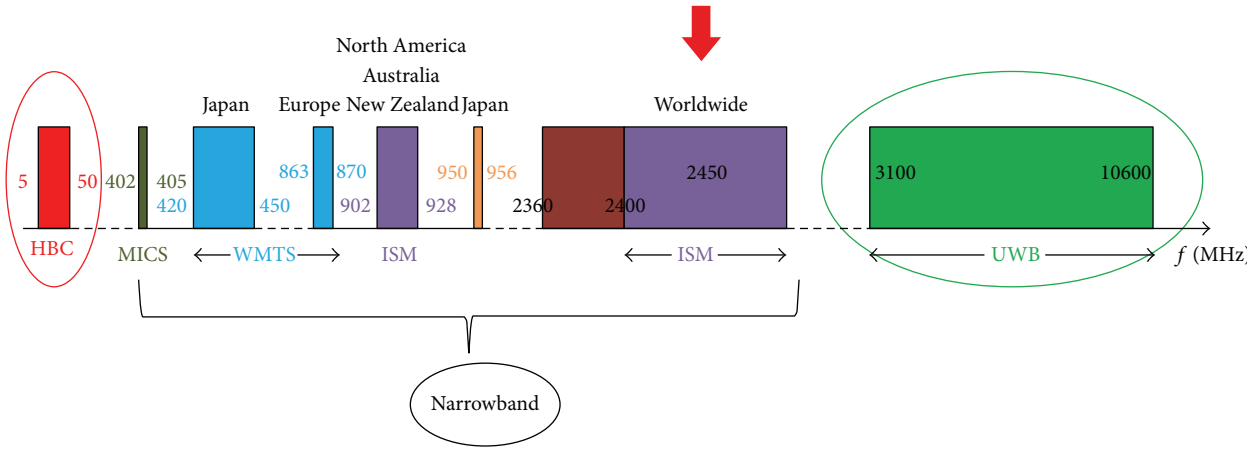


Figure 2: IEEE 802.15.6 frequency bands.

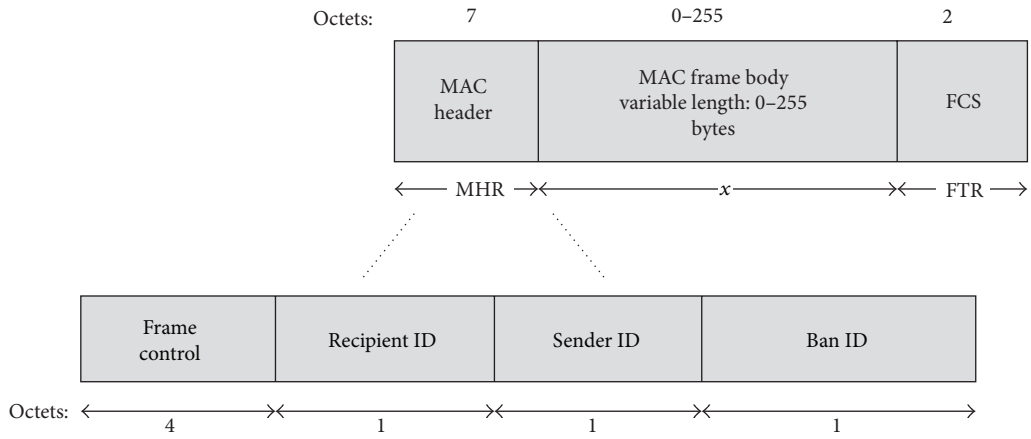
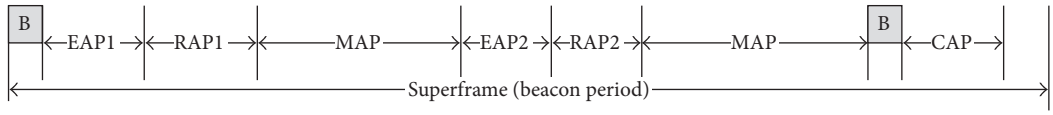
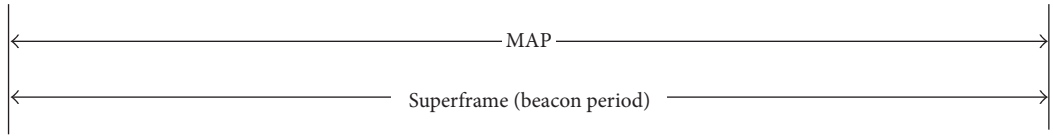


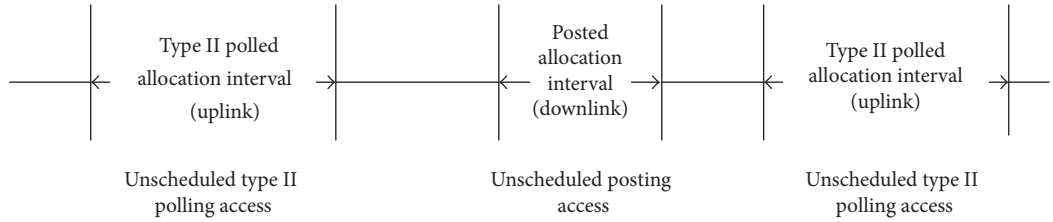
Figure 3: IEEE 802.15.6 MAC frame format.



(a) Beacon mode with superframe boundaries



(b) Nonbeacon mode with superframe boundaries



(c) Nonbeacon mode without superframe boundaries

Figure 4: IEEE 802.15.6 communication modes.

Table 1: Bounds for slotted-ALOHA and CSMA/CA protocols.

User Priorities	Slotted-ALOHA		CSMA/CA	
	CP_{max}	CP_{min}	CW_{min}	CW_{max}
0	0.125	0.0625	16	64
1	0.125	0.0937	16	32
2	0.25	0.0937	8	32
3	0.25	0.125	8	16
4	0.375	0.125	4	16
5	0.375	0.1875	4	8
6	0.5	0.1875	2	8
7	1	0.25	1	4

into a single EAP1 period and EAP2 and RAP2 into a signal EAP2 period. When using slotted ALOHA for high-priority traffic, EAP1 and EAP2 are not extended, but RAP1 and RAP2 are simply replaced by another EAP1 and EAP2 period. The following sections briefly describe the slotted ALOHA and CSMA/CA protocols.

(1) Slotted ALOHA Protocol. In the slotted ALOHA protocol, the nodes access the channel using predefined User Priorities (UPs), as given in Table 1. These priorities are used to classify the high- and low-priority traffic. Initially, the Collision Probability (CP) is selected according to the UPs. The nodes obtain contended allocation if $z \leq \mathrm{CP}$, where z is randomly selected from the interval [0, 1]. If the node fails to transmit, the CP remains unchanged into an odd number of failures and the node divides it equally for an even number of failures.

(2) CSMA/CA Protocol. In the CSMA/CA protocol, the node initially sets its backoff counter to a random integer that is uniformly distributed over the interval [1, Contention Window (CW)], where $\mathrm{CW} \in (\mathrm{CW}_{min}, \mathrm{CW}_{max})$. As given in Table 1, the values of CW_{min} and CW_{max} are selected according to the UPs. The high-priority traffic will have a small contention window compared to that of low-priority traffic, which increases the probability of accessing the channel to report emergency events. The node starts decrementing the backoff counter by one for each idle CSMA slot with a length equal to *pCSMASlotLength*. Particularly, the node considers a CSMA slot to be idle if it determines that the channel has been idle between the start of the CSMA slot and *pCCATime*. The node decreases the backoff counter *pCCATime* after the start of the CSMA slot. Once the backoff counter reaches zero, the node transmits the frame. If the channel is busy due to frame transmission, the node locks its backoff counter until the channel is idle. The CW is doubled for an even number of failures until it reaches CW_{max}. Figure 5 shows an example of the CSMA/CA protocol. As shown in the figure, the node unlocks the backoff counter in RAP1. However, the contention fails and the value of CW remains unchanged because CW does not change for an odd number of failures. In the following CAP period, the backoff counter is set to five; however, it is locked at two because the time between the end of the slot and the end of the CAP is not sufficient to accommodate the data frame transmission and Nominal Guard Time (GT_n). The backoff counter is then unlocked in the RAP2 period. This time, the value of CW is doubled because there is an even number of contention failures. The backoff counter is set to eight and is unlocked. Once the backoff counter reaches zero, the data are transmitted and the value of CW is set to CW_{max}.

We analysed the performance of an IEEE 802.15.6-based CSMA/CA for NB PHYs in terms of the theoretical throughput, delay and bandwidth efficiency, as presented in Table 2 [19–21]. The theoretical bounds are presented for No Acknowledgement (NACK) and Acknowledgement (ACK) policies. This analysis considers a single sender and single receiver with no collisions on the channel. The theoretical throughput and delay are bounded because of the additional overhead accumulated in each layer. These strict bounds cannot be achieved in a practical environment. However, these bounds can be used by the system designer for network provisioning and packet-size optimisation for different application scenarios.

2.3.2. Improvised and Unscheduled Access Mechanism. As discussed above, the hub may use improvised access to send poll or post commands without prereservation or advance notice in beacon or nonbeacon modes with superframe boundaries. These commands are used to initiate the transactions of one or more data frames by the nodes or hub outside the scheduled allocation interval. The polls are used to grant Type I or Type II polled allocation to the nodes, while the posts are used to send management frames. The Type I polled allocation starts after the duration of *pSIFS* and stops at the end of the allocated slot in the current superframe. Similarly, the Type II polled allocation starts after the duration of *pSIFS* and stops after all of the data frames are sent by the polled node. Figure 6 illustrates an example of immediate polled allocations.

The hub may also use an unscheduled access mechanism to obtain an unscheduled bilink allocation. The unscheduled bilink allocation may be (1) one-periodic, where frames are exchanged between the nodes and hub every superframe, or (2) multiple-periodic (m-periodic), where frames are exchanged every m superframes thus allowing the devices to sleep between m superframes. An m-periodic bilink allocation is suitable for low-duty cycle nodes because nodes in m-periodic allocation sleep between m superframes.

2.3.3. Scheduled and Scheduled-Polling Access Mechanisms. Unlike unscheduled allocation, the scheduled access mechanism is used to obtain scheduled uplink, downlink, and bilink allocations. In addition, the scheduled polling is used for polled and posted allocations. These allocations may be one-periodic or m-periodic; however, neither of these allocations is allowed in a single WBAN at the same time. The nodes consider the superframe periods (with allocated slots) as the wakeup periods. The uplink and downlink allocations are used to send management and data frames to and from the hub, respectively. Figure 7 illustrates an example of scheduled one-periodic allocations.

Table 2: Theoretical limits of IEEE 802.15.6 for NB PHYs.

420–450 MHz (PHY header rate = 57.5 kbps, PHY symbol rate = 187.5 kbps)							863–870 MHz (PHY header rate = 76.6 kbps, PHY symbol rate = 250 kbps)						
Data rate (kbps)	Maximum Throughput (kbps)		Theoretical delay (ms)		Bandwidth efficiency (%)		Data rate (kbps)	Maximum Throughput (kbps)		Theoretical delay (ms)		Bandwidth efficiency (%)	
	NACK	ACK	NACK	ACK	NACK	ACK		NACK	ACK	NACK	ACK	NACK	ACK
75.9	70.45	67.5	28.34	29.63	92.82	88.9	101.2	93.87	89.7	21.31	22.29	92.75	88.6
151.8	134.8	129	14.79	15.45	88.8	85.2	404.8	329.9	315.5	6.06	6.33	81.5	77.9
187.5	163.2	156	12.21	12.75	87.03	83.6	607.1	457.8	437.9	4.36	4.56	75.4	72.1

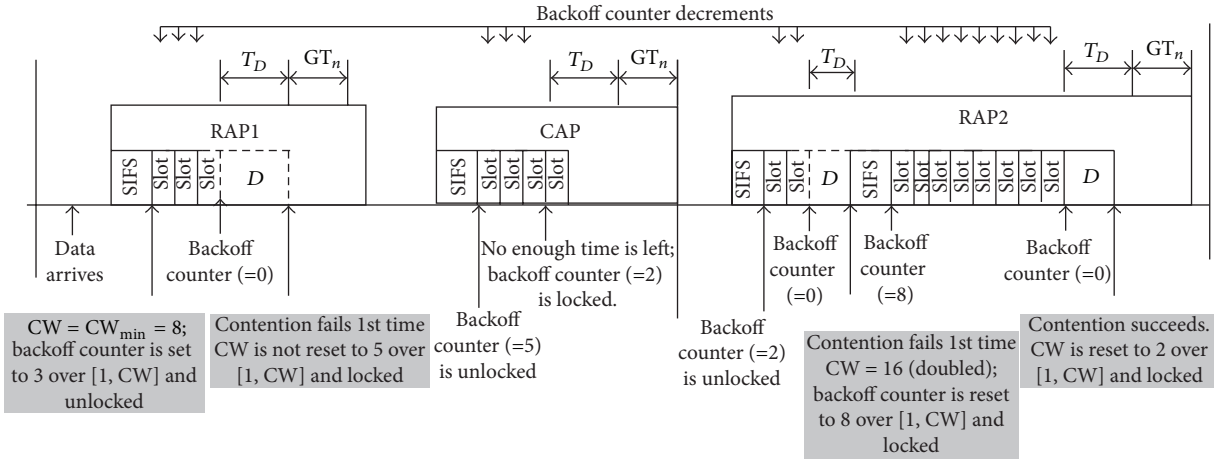


Figure 5: IEEE 802.15.6 CSMA/CA protocol: slot = CSMA slot SIFS = Psifs, D = frame transaction initiated by node 1 in a contended allocation (e.g., a data type frame and an I-Ack frame with pSIFS in between), T_D = time required to complete D, GT$_n$ = nominal guard time.

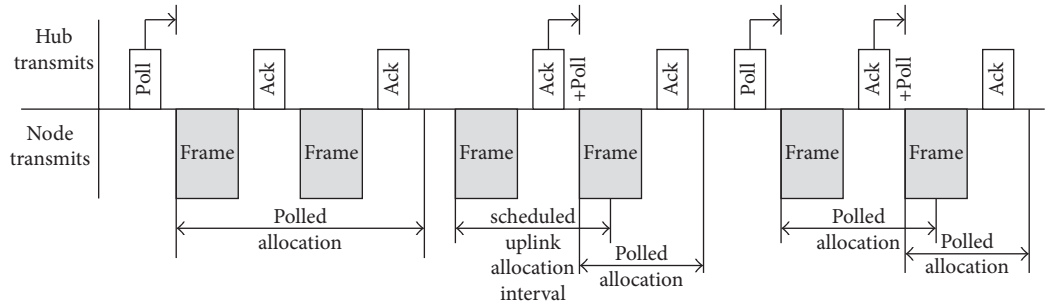


Figure 6: Immediate polled allocations.

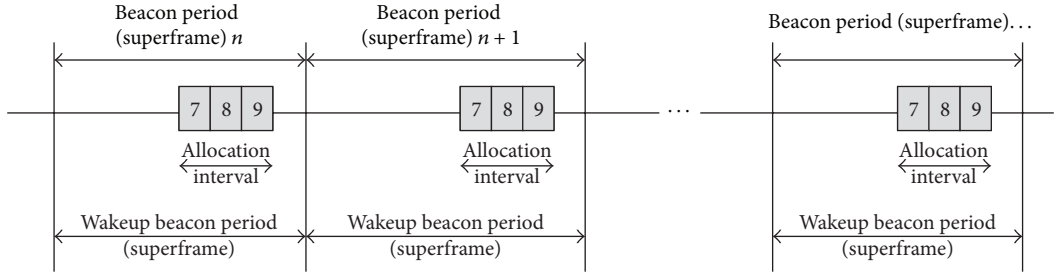


Figure 7: Scheduled one-periodic allocation.

Table 3: Transmission parameters in NB PHY.

Frequency band	Packet component	Modulation	Modulation order (M)	Symbol rate (R_s ksps)	Code rate BCH (n, k)	Spreading factor (SF)
402–405 MHz	PLCP header PSDU	π/M-DPSK	2 {2, 2, 4, 8}	187.5	(31, 19) (63, 51)	2 {2, 1, 1, 1}
420–450 MHz	PLCP header PSDU	GMSK	2	187.5	(31, 19) {(63, 51), (63, 51), 1}	2 {2, 1, 1}
863–870 MHz	PLCP header PSDU	π/M-DPSK	2 {2, 2, 4, 8}	250	(31, 19) (63, 51)	2 {2, 1, 1, 1}
902–928 MHz	PLCP header PSDU	π/M-DPSK	2 {2, 2, 4, 8}	250	(31, 19) (63, 51)	2 {2, 1, 1, 1}
950–958 MHz	PLCP header PSDU	π/M-DPSK	2 {2, 2, 4, 8}	250	(31, 19) (63, 51)	2 {2, 1, 1, 1}
2360–2400 MHz	PLCP header PSDU	π/M-DPSK	4 {2, 2, 2, 4}	600	(31, 19) (63, 51)	4 {4, 2, 1, 1}
2400–2483.5 MHz	PLCP header PSDU	π/M-DPSK	2 {2, 2, 2, 4}	600	(31, 19) (63, 51)	4 {4, 2, 1, 1}

3. IEEE 802.15.6 PHY Specifications

IEEE 802.15.6 supports three operational PHYs, two of which are mandatory and one of which is optional. The two mandatory PHYs are UWB and HBC PHYs, while the NB PHY is considered optional. In IEEE 802.15.6, PHY is responsible for (1) the activation and deactivation of the radio transceiver, (2) clear channel assessment, and (3) data reception and transmission. The following sections present the NB, HBC, and UWB PHY specifications of IEEE 802.15.6.

3.1. NB PHY Specifications

3.1.1. Operation Bands and Rate-Dependent Parameters. Table 3 summarises the supported frequency bands and corresponding PHY parameters, where WBAN devices must be able to support transmission on at least one of those frequency bands. Differential Phase Shift Keying (DPSK) is used except in the case of the 420 to 450 MHz frequency band, where Gaussian Minimum Shift Keying (GMSK) is employed. In some bands, several parameters have a set of values, where the underlined configurations in Table 3 are optional. Based on the table, the information data rate R_d is given by

$$R_d = \left(\frac{R_s \cdot N}{S} \times \frac{k}{n} \right) (\text{kbps}), \quad (1)$$

where R_s is the symbol rate, S is the spreading factor, k/n is the BCH coding rate, and M is the modulation order and is given by $M = 2^N$.

3.1.2. NB PHY PPDU Structure. The Physical-layer Protocol Data Unit (PPDU) encapsulates the Physical-layer Service Data Unit (PSDU) in its frame and appends several control fields that are used to synchronise the transmission and identify the transmission parameters. Figure 8 depicts the structure of the NB PPDU, and we identify the goal of each field in the following clauses.

PLCP Preamble. The preamble of the Physical-layer Convergence Protocol (PLCP) is a concatenation of two sequences. The first sequence has a length of 63 bits and is used for coarse time synchronisation, carrier-offset recovery, and packet detection. This sequence has two patterns; one pattern is used with odd-indexed channels, and the other is used with even-indexed channels. The second sequence has a fixed pattern length of 27 bits. It is appended to the first sequence and used for fine timing synchronisation.

PLCP Header. The PLCP header consists of several fields that convey the PHY parameters to the receiver, which is referred to as the PHY header and has 16 parity check bits appended. The details of these fields are as follows.

(1) Rate: three bits are used to indicate the information data rate computed using (1), which implicitly indicates the modulation, modulation order, code rate, and spreading factor.

(2) Length: eight bits are used to indicate the length of the MAC body in bytes (0–255 bytes).

(3) Burst mode: one bit is used to indicate the burst transmission.

(4) Scrambler seed: a 1-bit seed that identifies the initial state of the registers in the scrambler. It is initiated to zero and inverted after each PHY frame transmission.

(5) HCS: the Header Check Sequence, which is used for error detection, consists of a 4-bit Cyclic Redundancy Check (CRC-4) that is used to protect the PHY header.

(6) BCH parity check: the BCH field is computed for the concatenation of PHY header and HCS and is used for error correction. This check corrects up to two erroneous bits.

PSDU. As discussed above, the PSDU consists of a MAC header, a MAC frame body, and the FCS.

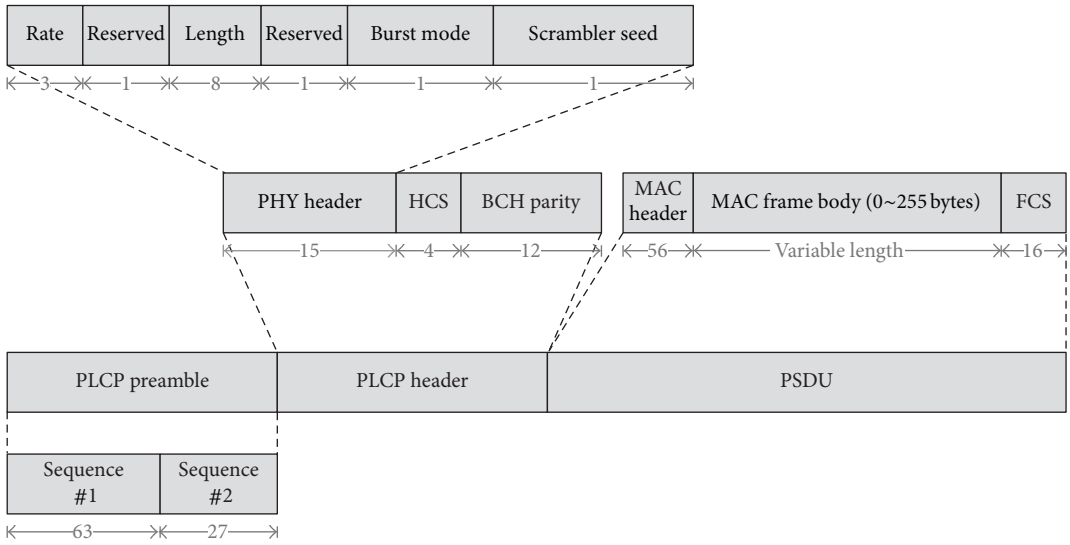


Figure 8: Standard PPDU structure for NB PHY (indicated lengths are in bits).

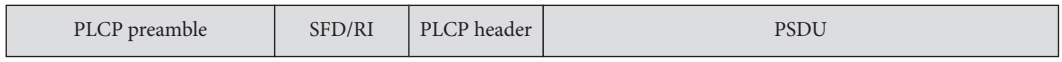


Figure 9: PPDU structure for HBC PHY.

3.1.3. Other NB PHY Specifications

Pulse Shaping. Square-root Raised Cosine (SRRC) is used as the shaping filter in all bands except in the 420 to 450 MHz band, where the Gaussian filter with a bandwidth-time product of 0.5 is used because the latter pulse shaping is recommended for use with GMSK.

Spreading. Bits are spread using simple repetition coding.

Transmit Mask. The transmitter should keep its out-of-band radiations to −20 dBr at most.

Power-On and Power-Off Ramp. The transmit power-on to power-off (or vice versa) ramp should take the duration of five symbols at most.

Receiver Sensitivity. The receiver sensitivity depends on the frequency band and transmission rate, and it ranges from −95 dBm (low frequency and low data rate) to −83 dBm.

Scrambler. The PSDU is whitened using a scrambler with a polynomial of order 14.

3.2. HBC PHY Specifications. HBC PHY uses Electric Field Communication (EFC) technology with the band of operation centred at 21 MHz. Similar to the NB PHY, the HBC packet structure encapsulates the PSDU in the packet after adding control bits and error correction and detection bits, as depicted in Figure 9. In the following, we introduce details about the packet structure.

PLCP Preamble. The initial preamble is generated as a 64-bit gold code sequence that is then repeated four times and spread using a Frequency Shift Code (FSC) with a Spreading Factor (SF) of eight to attain a chip rate of 42 Mcps, where cps denotes chips per second.

Start Frame Delimiter. The SFD indicates the starting point of the frame. Its sequence is generated using a 64-bit gold code generator, which is spread using an FSC with an SF of eight.

Rate Indicator. The SFD/RI field can be used to indicate the rate of transmission. In this case, the receiver does not need to read the PHY header to determine the data rate.

PHY Header. The PHY header is a 32-bit sequence, which is spread at a rate of 42 Mcps. The PHY header is composed of the following fields (we omit the fields already explained for the NB PHY).

(1) Pilot information: two bits indicate the length of the pilot insertion interval. The pilot sequence, which is the same as the SFD sequence, is periodically inserted in the PSDU to maintain synchronisation. If the PSDU is less than the pilot insertion interval, that is, short packet, the pilot is not needed.

(2) CRC8: CRC value is calculated over the PHY header to detect errors at the receiver side.

Scrambler. The PSDU is whitened using a scrambler with polynomial of order 32.

3.3. UWB Specifications. Compared to the two preceding PHY specifications, UWB PHY aims to achieve high performance, low complexity, and low power consumption. In addition, this specification is robust and in compliance with

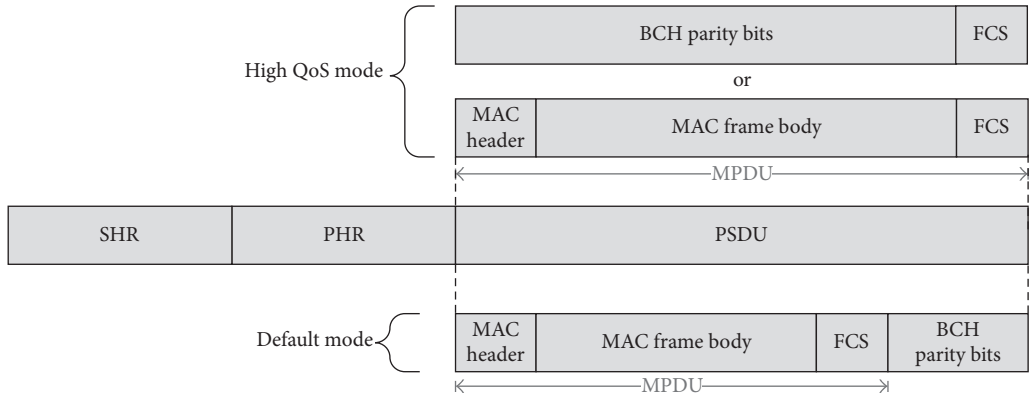


Figure 10: PPDU frame structure for UWB PHY.

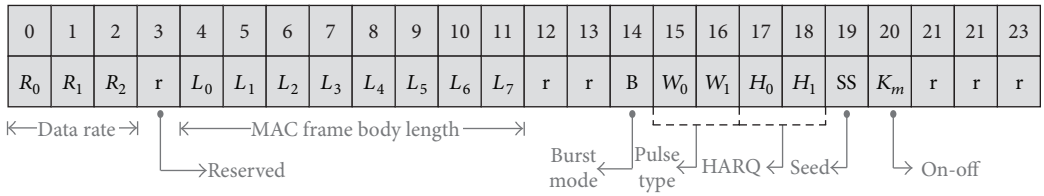


Figure 11: PHR frame structure.

the MICS power limits, which provide safe power levels for human body exposure. UBW PHY supports 11 channels; three in the low band (channels 0–2) and eight in the high band (channels 3–10), where the central frequencies range from 3,494.4 MHz for channel 0 to 9,984 MHz for channel 10, each with a 499.2 MHz bandwidth.

3.3.1. Supported Transceivers and Modes. UWB PHY supports Impulse Radio UWB (IR-UWB) and wideband Frequency Modulation UWB (FM-UWB) technologies. A hub can implement only one of these technologies, but a device can implement either IR- or FM-UWB or both technologies. Furthermore, UWB PHY supports two modes of operation, default mode and high Quality of Service (QoS) mode, where the latter is designated for high-priority medical applications and the default mode is used for medical and nonmedical applications.

3.3.2. Modulation. On-off modulation, Differential-BPSK (DBPSK) modulation, and Differential-QPSK (DQPSK) modulation schemes are supported for the IR-UWB, whereas Continuous Phase Binary FSK (CP-BFSK) is used for the FM-UWB, where the signal obtained from the CP-BFSK modulator is modulated again using wideband FM to create a constant-envelope UWB signal. The wideband FM increases the required transmission bandwidth by a factor of 256.

3.3.3. IR-UWB Symbol Structure. Each symbol time T_{sym} consists of an integer number of pulse waveform positions N_w, each of duration T_w. In DPSK and on-off modulation, the $N_w - 1$ and $((N_w/2) - 1)$ waveform positions, respectively, are used for time hopping. As such, UWN PHY supports the coexistence of multi-BANs.

3.3.4. UWB PHY Frame Format. The PPDU for a UWB frame consists of the Synchronisation Header (SHR), Physical-layer Header (PHR), and PSDU.

PSDU. As shown in Figure 10, the content of the PSDU depends on the operation mode; in the default mode operation, the PSDU is the concatenation of the MAC Protocol Data Unit (MPDU) and the BCH parity bits, whereas it consists of either the MPDU or BCH parity bits when operated in high QoS mode. The data bits of the MPDU are scrambled to randomise the data stream, block encoded, and interleaved to generate the PDSU for transmission.

BCH Encoder. The BCH (63, 51) and BCH (126, 63) are used in the default mode and high QoS mode, respectively. The BCH (126, 63) is used in accordance with the Hybrid Automatic Repeat Request (HARQ), which is explained below.

Bit Interleaving. Interleaving is applied to achieve robustness against error propagation; that is, multiple consecutive bits are erroneous at the receiver side. A simple modulus interleaver with a fixed size is used.

3.3.5. PHR Construction. The 24 PHR data fields shown in Figure 11 are appended with 4-bit CRC-4 ITU error detection before being encoded using BCH (40, 28) and are finally fed to the PPDU frame.

Data Rate (R_0 – R_2). These three bits specify the data rate, the symbol duration, the BCH coding rate, and other modulation-related parameters. In the case of IR-UWB, five sets of parameters are defined in the case of on-off modulation, and eight sets are defined when any of the

supported differential modulation schemes is used. In the case of FM-UWB, a single datum is defined by setting the three bits to zero, and the other options are reserved.

Pulse Shape (W_0 – W_1). UWB PHY supports three pulse shapes: chirp pulse, chaotic pulse, and short pulse shapes. The last option, that is, $W_0 = W_1 = 1$, is reserved.

Hybrid Automatic Repeat Request ($H_0 - H_1$). HARQ is used to permit the repetition of the transmission of certain packets in case an error occurs during the transmission. In UWB PHY, the maximum number of retransmissions is set to four. In the default mode, the BCH parity bits are appended to the MPDU to form the PSDU without requiring the HARQ technique. In the high QoS mode, the transmitter encodes the systematic data D (the MAC header and MAC frame body) to obtain the parity sequence P of the same size. Both sequences are saved at the transmitter. First, D is appended with its FCS to form the transmitted PSDU. If an error occurs and thus no ACK is received, P is appended with its FCS and transmitted. At the receiver, both D and P are used to recover the original data using BCH decoding. This process is repeated until either a successful decoding is achieved or the maximum number of retransmissions is exceeded. In this algorithm, $(H_0, H_1) = (0, 0)$, $(1, 0)$, $(0, 1)$, and $(1, 1)$ indicates that HARQ is disabled (default mode). BCH encoding is applied to D and P, and the algorithm sends D, D, and P.

Scrambler Seed (SS). This bit selects the initial state of the registers of the scrambler among the two available initial states.

Constellation Mapper for On-Off Modulation. The $K_m = 0$ and 1 refer to 16-ary (optional) and 2-ary (mandatory) waveform coding, respectively.

3.3.6. Synchronisation Header. The SHR consists of the preamble, which is used for timing synchronisation, packet detection, and carrier frequency offset recovery, and the start-of-frame delimiter (SFD), which is used for frame synchronisation.

Preamble. The preamble is a Kasami sequence of length 63, where eight sequences are available. The first four sequences are used for physical channels with an odd number, and the last four sequences are used for physical channels with an even number. The coordinator uses the preamble sequence with the minimum receiver power.

Start-of-Frame Delimiter. The SFD is the inversion of the Kasami sequence used ($0 \rightarrow 1$ and $1 \rightarrow 0$.) This choice leads to a minimised correlation between the SFD and preamble such that the detection of the SFD becomes more accurate.

4. IEEE 802.15.6 Security Specifications

The IEEE 802.15.6 supports three security levels with different security properties, protection levels, and frame formats.

Unsecured Communication Level. This level is the lowest level of security, where data are transmitted in unsecured frames. This level has no mechanisms for data authentication and integrity, confidentiality, or privacy protection.

Authentication Level. This level is the medium level of security, where data are transmitted in secured authentication with no encryption. This level does not support confidentiality or privacy.

Authentication and Encryption. This level is the highest level of security, where data are transmitted in authenticated and encrypted frames. This level provides solutions to all problems that are not covered by the lowest and medium security levels.

One of the security levels above is selected during the association process. A Master Key (MK) is activated for unicast secured communication. The MK may be preshared or established using unauthenticated association. Then, a Pairwise Temporal Key (PTK) is created for a single session. For multicast secured communication, a Group Temporal Key (GTK) is shared with the corresponding group using the unicast method. Figure 12 illustrates the process of activating MK and establishing PK for secured communication.

4.1. Security Association and Disassociation Procedure. IEEE 802.15.6 security protocols are generally based on the Diffie-Hellman key exchange, which employs the elliptic curve public key cryptography. The private keys used in the association and disassociation processes should be independent and unique 256-bit integers. The Cipher-based Message Authentication Code (CMAC) is used to derive the Key Message Authentication Codes (KMAC) and MK [22–24]. Initially, the node and hub have a preshared MK, which is used for the security association procedure. The node initiates the process by sending a security association frame request to the hub, as illustrated in Figure 13(a). The hub responds by either joining or aborting the association procedure. If the node receives a respond indicating an abortion, it stops the current association procedure. If the node receives a joining response, the preshared MK is activated and shared between the node and hub upon mutual agreement, which is then used to generate a PTK.

The disassociation procedure can be initiated either by the node or hub. As illustrated in Figure 13(b), the sender sends a security disassociation frame request and eventually removes the MK and corresponding PTK from its storage. When the recipient receives the request, it also removes the aforementioned key information from its storage.

4.2. PTK and GTK Procedures. After the MK is shared using the association procedure above, the node or hub steps towards generating a PTK. The node (or hub) sends a PTK frame request to the hub (or node). The recipient responds by either joining or aborting the procedure using the PTK field in the frame payload. The sender stops the procedure if a negative response is received; otherwise, it continues to send PTK frame requests to the recipient. The second PTK request is sent only after the successful verification of the PTK field in the frame payload. Once the second PTK request is received, the sender and recipient generate a new PTK.

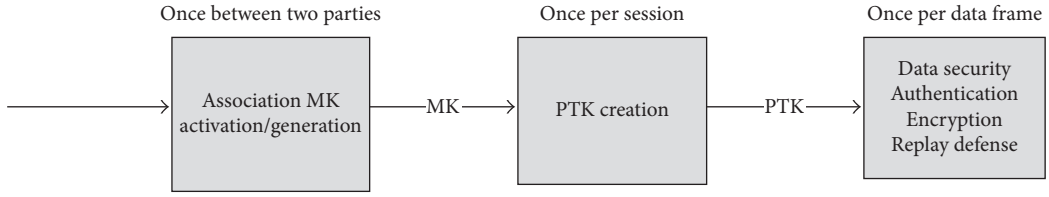
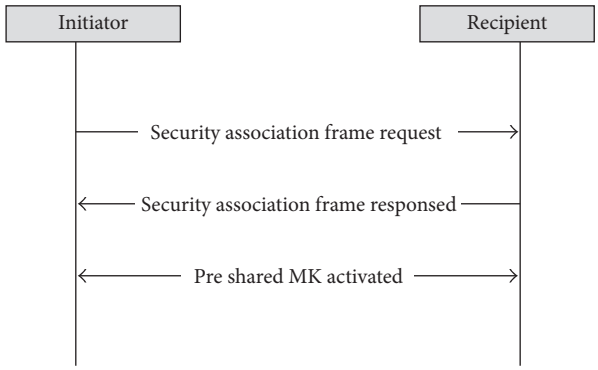
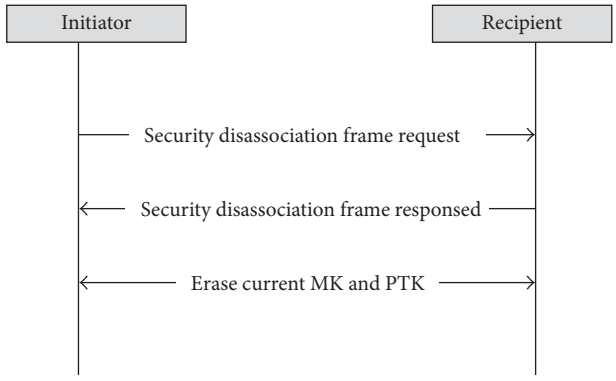


Figure 12: IEEE 802.15.6 security structure.



(a) Security association procedure of IEEE 802.15.6



(b) Security disassociation procedure of IEEE 802.15.6

Figure 13: Security association and disassociating procedures of IEEE 802.15.6.

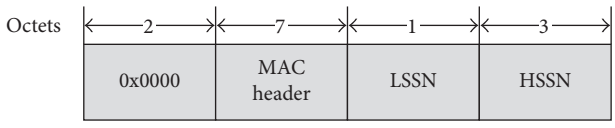


Figure 14: IEEE 802.15.6 nonce format.

The GTK is distributed among the nodes using the PTK. The hub sends a GTK to the node to multicast secured frames to other nodes.

4.3. Message Security. The frames can be transmitted in both secured and unsecured communication modes. The nodes that do not require security receive all frames including beacons without validating the security information. The secured frames are authenticated and encrypted or decrypted using the AES-128 Counter (CCM) [25–27] mode. As illustrated in Figure 14, a 13-octet nonce is required for each instance of CCM frame authentication and encryption or decryption.

The Low-order Security Sequence Number (LSSN) is set to zero if the frame is secured with a new PTK or GTK and is incremented by one if the frame is a retransmission of the last frame. The High-order Security Sequence Number (HSSN) is set to zero if the frame is secured with a PTK. The HSSN is incremented by one if the security sequence number of the current frame is less than that of the last frame.

5. Conclusions

This paper presented the most important features of the IEEE 802.15.6 standard. A deep explanation of MAC, PHY, and security specifications of the standard was presented. Different communication modes and access mechanisms were explained. The NB, HBC, and UWB PHY specifications were reviewed in terms of frame structure, modulation, and other key parameters. In addition, the security services, including key generation and message security, were discussed. We believe that this paper could be used to quickly understand the key features of the standard and to analyse its potential for different applications.

Abbreviations

MAC:	Medium Access Control
PHY:	Physical layers
NB:	Narrowband PHY
UWB:	Ultra-wideband PHY
HBC:	Human Body Communications
WBANs:	Wireless Body Area Networks
CE:	Consumer Electronics
MICS:	Medical Implant Communications Service
WMTS:	Wireless Medical Telemetry Services
ISM:	Industrial, Scientific, and Medical
FCS:	Frame Check Sequence
ID:	Identification
MIC:	Message Integrity Code
EAP:	Exclusive Access Phases
RAP:	Random Access Phases
MAP:	Managed Access Phase
CAP:	Contention Access Phase
CSMA/CA:	Carrier Sensor Multiple Access/Collision Avoidance
Ups:	User Priorities
CP:	Collision Probability

NACK: No Acknowledgement
ACK: Acknowledgement
DPSK: Differential Phase Shift Keying
GMSK: Gaussian Minimum Shift Keying
PPDU: Physical-layer Protocol Data Unit
PSDU: Physical-layer Service Data Unit
PLCP: Physical-layer Convergence Protocol
CRC-4: 4-bit Cyclic Redundancy Check
HCS: Header Check Sequence
SRRC: Square-root Raised Cosine
EFC: Electric Field Communication
FSC: Frequency Shift Code
SF: Spreading Factor
IR-UWB: Impulse Radio UWB
FM-UWB: Frequency Modulation UWB
QoS: Quality of Service
DBPSK: Differential-BPSK
DQPSK: Differential-QPSK
CP-BFSK: Continuous Phase Binary FSK
SHR: Synchronization Header
MPDU: MAC Protocol Data Unit
HARQ: Hybrid automatic repeat request
SS: Scrambler seed
SFD: Start-of-frame Delimiter
MK: Master Key
PTK: Pairwise Temporal Key
GTK: Group Temporal Key
KMAC: Key Message Authentication Codes
CCM: AES-128 Counter
LSSN: Low-order Security Sequence Number
HSSN: High-order Security Sequence Number.

Authors' Contribution

All of the authors contributed equally to this paper.

Acknowledgment

This work is supported by the Research Center of College of Computer and Information Sciences, King Saud University. The authors are grateful for this support.

References

[1] S. Ullah, H. Higgins, B. Braem et al., "A comprehensive survey of wireless body area networks—on PHY, MAC, and network layers solutions," *Journal of Medical Systems*, vol. 36, no. 3, pp. 1065–1094, 2012.

[2] M. Chen, "Body area networks: a survey," *Mobile Networks and Applications*, vol. 16, no. 2, pp. 171–193, 2011.

[3] B. Latré, B. Braem, I. Moerman, C. Blondia, and P. Demeester, "A survey on wireless body area networks," *Wireless Networks*, vol. 17, no. 1, pp. 1–18, 2011.

[4] 2012, http://www.ieee802.org/15/pub/TG6.html.

[5] T. Baykas, C. S. Sum, Z. Lan et al., "IEEE 802.15.3c: the first IEEE wireless standard for data rates over 1 Gb/s," *IEEE Communications Magazine*, vol. 49, no. 7, pp. 114–121, 2011.

[6] E. Karapistoli, F. N. Pavlidou, I. Gragopoulos, and I. Tsetsinas, "An overview of the IEEE 802.15.4a standard," *IEEE Communications Magazine*, vol. 48, no. 1, pp. 47–53, 2010.

[7] R. Lampe, R. Hach, and L. Menzer, "Chirp spread spectrum (CSS) PHY presentation for 802. 15. 4a, IEEE P802. 15," Working Group for Wireless Personal Area Networks (WPAN), 2004.

[8] J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile, "IEEE 802.15.4: a developing standard for low-power low-cost wireless personal area networks," *IEEE Network*, vol. 15, no. 5, pp. 12–19, 2001.

[9] IEEE WLAN, 2012, http://www.ieee802.org/11/.

[10] IEEE WPAN Task Group 1, 2012, http://www.ieee802.org/15/pub/TG1.html.

[11] *IEEE Std. 802. 15. 4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Data Rate Wireless Personal Area Networks (WPAN)*, IEEE, Piscataway, NJ, USA, 2006.

[12] "IEEE P802. 15. 6, Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs) used in or around a body," 2012.

[13] M. Hernandez and R. Miura, "Coexistence of IEEE Std 802. 15. 6TM-2012 UWB-PHY with other UWB systems," in *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB)*, pp. 46–50, September 2012.

[14] A. W. Astrin, H. B. Li, and R. Kohno, "Standardization for body area networks," *IEICE Transactions on Communications*, vol. 92, no. 2, pp. 366–372, 2009.

[15] H. B. Li, K. Takizawa, and R. Kohno, "Trends and standardization of body area network (BAN) for medical healthcare," in *Proceedings of the 1st European Wireless Technology Conference (EuWiT' 08)*, pp. 1–4, Amsterdam, The Netherlands, October 2008.

[16] S. Rashwand, J. Mišić, and H. Khazaei, "IEEE 802.15.6 under saturation: some problems to be expected," *Journal of Communications and Networks*, vol. 13, no. 2, pp. 142–148, 2011.

[17] S. Rashwand and J. V. Misic, "Effects of access phases lengths on performance of IEEE 802. 15. 6 CSMA/CA," *Computer Networks*, vol. 56, no. 12, pp. 2832–2846, 2012.

[18] S. Rashwand and J. Mišić, "Performance evaluation of IEEE 802.15.6 under non-saturation condition," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM' 11)*, pp. 1–6, Houston, Tex, USA, December 2011.

[19] S. Ullah and K. S. Kwak, "Throughput and delay limits of IEEE 802.15.6," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC' 11)*, pp. 174–178, Cancun, Mexico, March 2011.

[20] S. Ullah, M. Chen, and K. S. Kwak, "Throughput and delay analysis of IEEE 802. 15. 6-based CSMA/CA protocol," *Journal of Medical System*, vol. 36, no. 6, pp. 3875–3891, 2012.

[21] K. S. Kwak, S. Ullah, and N. Ullah, "An overview of IEEE 802.15.6 standard," in *Proceedings of the 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL' 10)*, pp. 1–6, Rome, Italy, November 2010.

[22] NIST Special Publication 800-38B, 2005.

[23] H. E. Michail, A. P. Kakarountas, G. Selimis, and C. E. Goutis, "Throughput optimization of the cipher message authentication code," in *Proceedings of the15th International Conference on Digital Signal Processing (DSP' 07)*, pp. 495–498, Cardiff, UK, July 2007.

[24] A. A. Adekunle and S. R. Woodhead, "Zone based systems design framework for the realisation of efficient block cipher

based message authentication code algorithms," in *Proceedings of the 5th International Conference on Availability, Reliability, and Security (ARES' 10)*, pp. 216–221, Krakow, Poland, February 2010.

[25] A. Astarloa, A. Zuloaga, J. Lázaro, J. Jiménez, and C. Cuadrado, "Scalable 128-bit AES-CM crypto-core reconfigurable implementation for secure communications," in *Proceedings of the Applied Electronics International Conference (AE' 09)*, pp. 37–42, Pilsen, Czech Republic, September 2009.

[26] M. K. Khan and J. Zhang, "Improving the security of 'a flexible biometrics remote user authentication scheme'," *Computer Standards and Interfaces*, vol. 29, no. 1, pp. 82–85, 2007.

[27] NIST Special Publication 800-38C, 2004.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 427275, 12 pages
http://dx.doi.org/10.1155/2013/427275

Research Article

Precision-Enhanced and Encryption-Mixed Privacy-Preserving Data Aggregation in Wireless Sensor Networks

Geng Yang,[1] Sen Li,[1] Xiaolong Xu,[1] Hua Dai,[1] and Zhen Yang[2]

[1] *Jiangsu High Technology Research Key Lab for WSNs, College of Computer Science & Technology, Nanjing University of Posts & Telecommunications, Nanjing 210046, China*
[2] *Key Lab of Broadband Wireless Communication and Sensor, Network Technology of Ministry of Education, Nanjing University of Posts & Telecommunications, Nanjing 210046, China*

Correspondence should be addressed to Geng Yang; yangg@njupt.edu.cn

Received 18 January 2013; Revised 29 March 2013; Accepted 30 March 2013

Academic Editor: Muhammad Khurram Khan

Security is always a hot topic in wireless sensor networks (WSNs). Privacy-preserving data aggregation has emerged as an important concern in designing data aggregation algorithm. This paper proposes a *precision-enhanced and encryption-mixed privacy-preserving data aggregation (PEPDA)*. The objective is to reduce collision during data transmission and energy consumption and to compensate loss caused by the collision. Based on the *Slice-Mix-AggRegaTe (SMART)* scheme, it optimizes data slicing by using small data packet, node classifying, and positive and negative data slicing techniques. It also describes a randomized time slot and a data compensation algorithm. Theoretical analysis and simulation show that *PEPDA* demonstrates a good performance in terms of accuracy, complexity, and security.

1. Introduction

Wireless sensor network (WSN) has received considerable attention during last decade. It has been developed for a wide variety of applications, including military sensing and tracking, environment and security monitoring, and equipment and human monitoring and tracking. Sensor networks usually consist of a large number of ultrasmall autonomous devices. Each device, called a node, is battery powered and equipped with integrated sensors, digital signal processors (DSPs), and radio frequency (RF) circuits. Because of special characteristics and limitations of wireless sensor networks, we face an important challenge in security issue, particularly for the applications where wireless sensor networks are developed in a hostile environment or used for some crucial purposes. For example, an adversary can easily listen to the traffic and mislead communications between nodes. Usually, one of the objectives to develop a sensor network is to collect data. We have therefore to establish a secure network and data aggregation mechanism, together with designing secure protocols to deal with problems about key agreement and encryption in communications and to develop privacy-preserving data aggregation algorithms.

Sensor nodes collect data from where they are deployed and forward the corresponding data to sink node. If some sensors are compromised, the aggregated result will be ill-performed; Chan et al. [1] and Yang et al. [2] have introduced intrusion detection to identify ill-performed aggregation. These are passive privacy-preserving schemes. Moreover, some positive privacy-preserving schemes then are proposed by using cryptographic mechanism to establish secure communication links. A key predistribution scheme was first presented by Eschenauer and Gligor in [3], and a series of improved key distribution schemes [4–6] were described after that. The predistribution keys can be used to construct a hop-by-hop secure data aggregation algorithm. It is a simple and effective way to employ the encryption in data aggregation. However, the encryption and decryption operations have to be executed at each node. Therefore, data aggregating cost is relatively high. In order to get efficiency in privacy-preserving data aggregation, homomorphic encryption was introduced to construct an end-to-end secure

data aggregation algorithm. This technique allows arithmetic operations to be performed on ciphertext directly. Note that the schemes using key distribution can ensure data not to be revealed by attackers from outside of network. However, a more stringent scenario may ask for guarantees of in-network confidentiality, which means that individual sensitive data should not be disclosed to any node in the network, including parent node or neighboring node. Some approaches are presented in [7–9] to address these issues. Meanwhile, a typical scheme, called SMART, is proposed by He et al. in [7], which slices individual sensitive data into a set of pieces and sends them to corresponding associated nodes. The SMART scheme guarantees privacy-preserving against attacks from outside and inside of a network by using encryption with the predistributed keys and slicing the data, respectively. It has attracted much more attention in research of privacy-preserving data aggregation.

The objective of this paper is to evaluate both security vulnerability and efficiency in data aggregating schemes, particularly for the SMART scheme, and to propose a novel optimal approach, because efficiency and privacy are two important factors considered in designing data aggregation algorithm. The network's whole lifetime is tied up with node's individual energy consumptions which are spent on processing instructions, computations of CPU, send and receive operations, and so forth. Based on SMART scheme, we propose a PEPDA scheme by optimizing some parameters to reduce data collision, data loss and overhead, then to prolong the lifetime of a WSN. Compared with SMART scheme, the proposed PEPDA scheme demonstrates better performances in terms of piece accepting rate, aggregation accuracy, energy consumption, and privacy-preserving efficacy.

The rest of this paper is organized as follows. Section 2 gives a summary of related work. Section 3 evaluates limitations in SMART scheme and introduces our improvement assumptions. Section 4 describes the PEPDA approach with five optimizing factors. Section 5 provides detailed PEPDA protocol. Section 6 analyzes performance of PEPDA scheme. Section 7 gives conclusions and sketches some future work.

2. Related Work

Security of data aggregation in WSNs has been investigated during last decade. Several references give a review about it [10]. Obviously, it is a good way to use cryptography to protect privacy of data.

Privacy-preserving data aggregation schemes using cryptographic mechanisms can be classified into two types: end-to-end encryption scheme and hop-by-hop encryption scheme [2, 7, 8, 11]. The end-to-end encryption scheme aims to establish secure link between base station (BS) and individual sensor node. Sensitive data are encrypted before forwarding upstream; BS then extracts original data using agreed key with each node, making intermediate nodes transparent during data transmission process. However, end-to-end encryption without aggregation is very power-consuming, because each encrypted data is transmitted to BS directly. Along with the fact that nodes closer to BS consume more energy as more data pass through them, the efficiency of end-to-end encryption without aggregation is debatable. To tackle this problem, homomorphic encryption technique is introduced by Castelluccia et al. in [12] and de Cristofaro in [13] to achieve in-network aggregation with end-to-end encryption. Some schemes were described to deal with addition operations in data aggregation with homomorphic encryption, such as to find sum or average value. Homomorphic encryption makes it possible to aggregate data without doing encryption and decryption at intermediate nodes. However, it is not easy to find out operation satisfying the homomorphic properties.

In hop-by-hop encryption scheme, upon receiving an aggregated data, the node decrypts it, aggregates with its own data, encrypts the newly aggregated data, and then forwards upstream. The encrypt and decrypt operations are performed by using certain key distribution scheme. Obviously, hop-by-hop encryption scheme is not an efficient design due to frequent intermediate encrypt and decrypt operations which brings about extra energy consumption and computational delay. Moreover, underlying privacy vulnerability is exposed when decrypted data are eavesdropped. Particularly, we face a challenge from inner attack. Piece slicing technique in SMART is a solution to this problem. In-network aggregation with end-to-end encryption scheme, on the contrary, provides better efficiency and does not have to worry about privacy vulnerability during intermediate node aggregation. Along with the attractive advantages, in-network aggregation with end-to-end encryption scheme designs however have to deal with certain problems in key distribution phase, which are elaborated by Feng et al. in [14].

Privacy-preserving data aggregation (PDA) scheme presented by He et al. in [7] consists of two schemes: cluster-based private data aggregation (CPDA) and SMART. Privacy performance is improved in these two schemes. However, neither of them is efficient. The former one is of computational complexity and big computational burden. Limitations of the latter scheme will be investigated in Section 3.

Some improved schemes are proposed based on the PDA scheme. Yang et al. present an energy-saving and privacy-preserving data aggregation (ESPART) scheme [8], which shows a good performance in both energy consumption and privacy-preserving efficacy. As a result, the lifetime of network could be prolonged. Work presented in [11] introduces a scheme which applies the additive property of complex numbers in order to combine sensor data and preserve data privacy during transmissions to the query server. The performance evaluation shows that it is more efficient than the PDA scheme in terms of both communication and computation costs. Work presented in [9] specializes in nonlinear aggregation functions instead of traditional additive function. The presented K-indistinguishable privacy-preserving data aggregation (KIPDA) scheme achieves the goal of privacy-preserving upon MAX and MIN aggregation functions by obfuscating data being forwarded. Zhang et al. proposed schemes to support both additive aggregation functions and nonadditive ones such as Max/Min, Median, and Histogram at the sacrifice in data accuracy [15]. A formal treatment to the security of concealed data aggregation (CDA) and

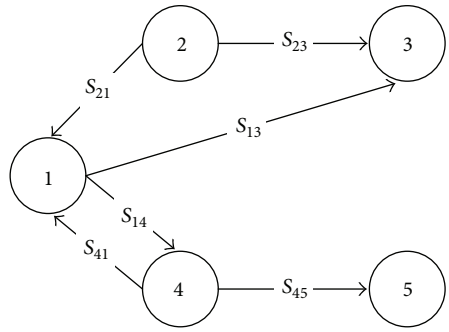


Figure 1: An example of slicing and forwarding in SMART.

the more general private data aggregation (PDA) is given in [16]. It analyzed security by comparing with SMART scheme. Despite the existence of schemes for privacy-preserving data aggregation, a rigorous analysis and optimization for SMART are still missing in the literature.

3. Background and Assumptions

3.1. SMART Scheme. Before introducing our proposal, we make a review about SMART scheme presented in [7], which consists of three steps.

Step 1 (Slicing). Each sensor node i randomly selects a set S_i of sensor nodes, say J, within h hops, and slices data d_i randomly into J pieces. One of the J pieces is kept at node i itself; the remaining $J-1$ pieces are encrypted by using their shared keys and sent to nodes randomly selected from the node set S_i.

Step 2 (Mixing). When a node j receives an encrypted slice, it decrypts the data using the shared key with the sender. Then, it sums up all the received slices.

Step 3 (Aggregating). All nodes aggregate the data according to the TAG protocol in [17] and send the result to the query server.

3.2. Analysis of SMART Scheme. This subsection evaluates SMART scheme in terms of collision and *slice failed node* (SFN). Assume that each node slices data into 3 pieces and forwards 2 of them to neighbors shown in Figure 1. We denote S_{ij} as a piece sent from node i to node j. If S_{14} and S_{41} are forwarded at the same time, a collision happened between node 1 and node 4. As a result, it influences aggregation result. The worst situation is that each piece collides with another. In fact, the aggregation accuracy is around 40% for $J = 3$, because the piece accepting rate is only about a quarter. This inspires us to design new accuracy oriented privacy-preserving scheme.

Another problem is the SFN. In SMART scheme $J-1$ pieces are encrypted and sent to nodes randomly selected from node set S_i. Unfortunately, for some nodes the number of nodes in the set S_i is less than $J-1$. Therefore, the encrypted pieces could not be sent to neighbors, which influence privacy-preserving efficacy in SMART. We define these nodes as SFN. For example, for a network with size $N = 600$, hop size $h = 1$, and slicing number $J = 3$, there exist averagely 24 nodes in SFN. If the value $|S_i|$ of the node i is lower than 2, it could not find enough destination nodes to send its corresponding 2 sliced pieces.

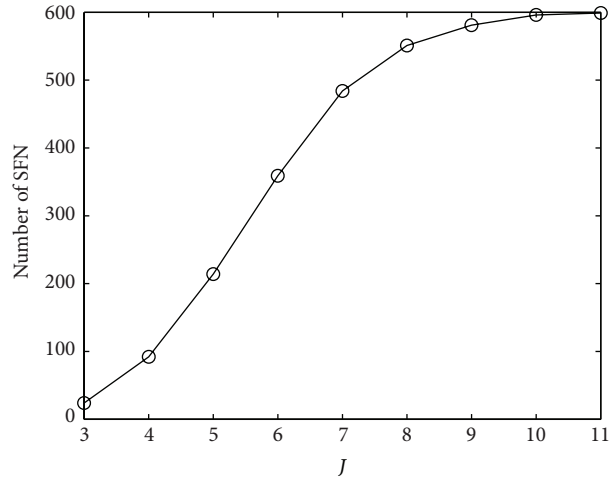


Figure 2: The relation between SFN and J.

Figure 2 shows the variation of SFN with the slicing number J. The number of SFN increases with the increase of slicing number J. This implies that the more pieces a node goes to slice, the more difficult it is to find enough corresponding destination nodes to send. One reason is that communication range of a sensor node is limited. Figure 2 also reveals that almost all nodes become SFN when slicing size J tends to 11, which indicates that SMART degenerates into TAG gradually [17].

The above analyses inspire us to optimize SMART scheme with some factors to reduce the collision rate and to increase data aggregation accuracy. Five factors will be evaluated, respectively, which are shown in Figure 3. In order to reduce collision rate, a randomized time slot and node choosing technique are developed, while to reduce collision loss, small data packet, positive and negative piece slicing, and compensation methods are presented. All are discussed in Sections 4 and 5.

4. Overview of Proposed Approach

4.1. System Model. Consider a network with N nodes. Each node is marked from 1 to N, and the ID of sink is 1. Assume that each node collects body temperature, which fluctuates from 35°C to 43°C, denote D_i as collected reading of node i, and the lower-bound and upper-bound of D_i are 35 and 43, respectively. There are many types of aggregation functions such as SUM, MAX, and AVERAGE. Here we only deal with SUM aggregation function $y(t) = \sum_{i=1}^{N} f(D_i(t))$ as other functions could be somehow simplified into SUM model [12]. The aggregation accuracy is defined as

$$p_a = \frac{D_1''}{\sum_{i=1}^{N} D_i}, \tag{1}$$

where D_1'' represents the final aggregation result in sink node.

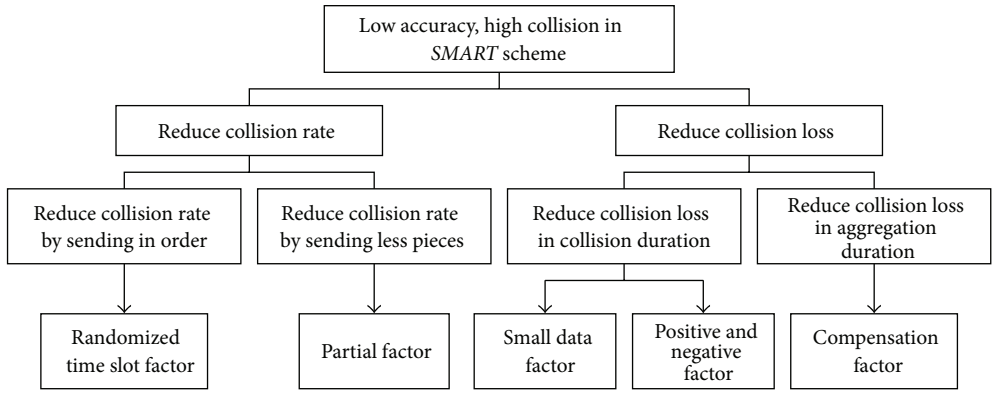


Figure 3: Improvement outline.

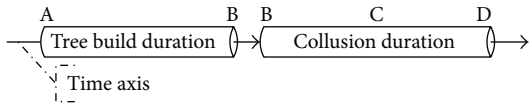


Figure 4: Piece forwarding moment diagram.

Epoch duration presents the amount of time in data aggregation process, which is divided into four intervals: tree formatting phase, collusion phase, postback phase, and aggregation phase. Tree formatting phase expresses the time interval assigned for N nodes to establish a treelike hierarchical structure, collusion phase is a time interval for nodes to send sliced pieces to others, postback phase describes the time interval for nodes to send acknowledgment, and aggregation phase specifies the time interval for network to aggregate according to certain aggregation protocol.

4.2. The Proposed Approach. This subsection describes how to optimize SMART scheme. Detailed algorithms will be presented in Section 5.

4.2.1. Randomized Time Slot Factor. In SMART scheme, each node slices data into J pieces and sends them to neighbors. In order to reduce collision rate, a random sending time schedule is used during collusion phase, instead of spontaneously sending sliced pieces at the same time. Figure 4 demonstrates a piece forwarding time diagram. The first phase from A to B is tree build duration, while the second phase from B to D is collusion duration. Assume that slice number J is 3, therefore, every node has 2 pieces to send. The piece forwarding moments in SMART are B and C, respectively, (C is the midpoint of the line BD). In PEPDA scheme, the forwarding times are set randomly in time slots BC and CD, respectively. This forwarding mechanism demonstrates a good performance in reducing data collision which will be shown later.

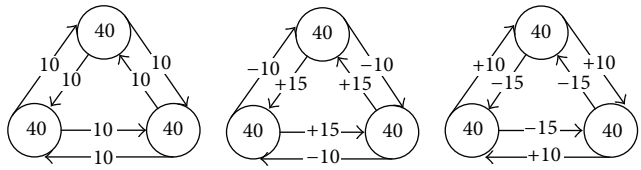
4.2.2. Partial Factor. Note that the node in the set of SFN cannot find enough neighbor nodes to send $J - 1$ pieces. Therefore, there is at least one edge used to transmit twice the pieces. This is one of the reasons to cause collision. As a remedy, we divide nodes into two subsets T and F based on the condition $|S_i| \geq J - 1$, where $i = 1, \ldots, N$. The node set F contains all SFN nodes, while the rest is in the node set T. Only nodes from the set T participate in piece slicing and mixing. Therefore, communication overhead is cut down, and then collision rate and energy consumption are reduced.

4.2.3. Small Data Factor. In SMART scheme, $J - 1$ pieces will be forwarded to neighbors. There is the possibility that large pieces are sent out while a small piece is kept by the node itself; under this circumstances, if collision occurs, most part of the data will be lost. Therefore, aggregation accuracy will be influenced. In order to improve the performance of SMART scheme, we define a small data factor L and make small fragments to be sent to neighbors. A detailed algorithm is given in the next section.

4.2.4. Positive and Negative Factor. Figure 5 illustrates three special slicing cases. Assume that slicing number J is 3 and all of the corresponding sent pieces are dropped as a result of collision. Data D_i $(i = 1, 2, 3)$ are 10. The numerical number on the line presents data sent or received by a node. According to the definition of the aggregation accuracy, we have the following results:

$$p_{\text{case 1}} = \frac{(40 - 10 - 10) \times 3}{40 \times 3} = 0.5,$$
$$p_{\text{case 2}} = \frac{[40 - 15 - (-10)] \times 3}{40 \times 3} = 0.875, \quad (2)$$
$$p_{\text{case 3}} = \frac{[40 - (-15) - 10] \times 3}{40 \times 3} > 1.$$

From the analysis of case 1 and case 2, we notice that aggregation accuracy can be improved if a negative piece is sent. One reason is that using a negative piece increases proportion of the data kept by the node itself and decreases the influence of data loss caused by collision. As it is shown in case 2, $\Sigma_m \text{Positive Piece}_m \geq \Sigma_n |\text{Negative Piece}_n|$, aggregation accuracy will be improved. Otherwise, aggregation accuracy will be distorted (see case 3). Such a negative piece technique is used in PEPDA scheme.



(a) case 1 (b) case 2 (c) case 3

Figure 5: Three slicing cases.

4.2.5. Compensation Factor. During the collusion phase, some pieces get lost because of collision, which finally influences the aggregation result at the sink node. If a node knows whether a piece is received by a neighbor successfully or has the loss rate, it can compensate for aggregating data and forward the result upstream during the data aggregation phase. This process needs to solve two problems: one is how to get the loss rate; the other is how to calculate the compensation. In PEPDA scheme, an ACK message will be sent to the neighbor to get the loss rate, and also an algorithm is presented to determine the compensation in Section 5.

5. Algorithms and Their Property

This section describes details of PEPDA scheme. The randomized time slot factor, the partial factor, the positive and negative factor, and the small data factor are used in the collusion phase, while the compensation factor is taken in the postback phase and the aggregation phase.

5.1. Tree Formatting Phase. An aggregation tree is constructed in the following way according to the standard aggregation protocol TAG [17].

Step 1. Sink node 1 marks its tree level 0 and broadcasts a *Hello* message which contains its level information L_v.

Step 2. On receiving a *Hello* message, the node drops the message if it is already in the aggregation tree. Otherwise the node extracts L_v value from the packet and marks its tree level $L_v + 1$. Accordingly, the source node in the packet becomes parent node of this node. Then the node continues to broadcast *Hello* message containing level information L_v of its own.

Step 3. Loop Step 2 until all nodes are added to the aggregation tree.

As aggregation tree is being constructed, are the node set S_i $(i = 1, \ldots, N)$ is established in the following way.

Upon receiving a *Hello* packet during the tree formatting phase, the node i records the source address number of this *Hello* packet into a memory space such as a neighboring table. The node set S_i is then selected from such kind of neighboring table according to a fixed value (say $|S_i| = J$).

This phase establishes also shared keys used in encryption/decryption. We refer to the existing random key distribution mechanism proposed in [3].

5.2. Collusion Phase. This phase in PEPDA scheme is quite different from that of SMART scheme. In the tree formatting phase, the node set S_i is set up. All nodes are then divided into two subsets according to the condition $|S_i| \geq J - 1$. Nodes satisfying or against the condition $|S_i| \geq J - 1$ are classified into node sets T or F, respectively. Each node i $(i = 1, \ldots, |T|)$ from the node set T slices data D_i into J pieces. One of the J pieces is kept at the node i itself; the remaining $J - 1$ pieces are encrypted and sent to nodes randomly selected from the node set S_i. When a node receives an encrypted piece, it decrypts the data using its shared key with the sender. Meanwhile, the node extracts the source address of this piece packet and adds it to an ACK forwarding table which is used for forwarding ACK messages later. The encrypting and decrypting operations are performed with the keys distributed by a random key distribution mechanism mentioned above.

Based on the technique of the positive and negative factor and the small data factor, the sent pieces are calculated in the following way.

Assume that the $J - 1$ pieces to be sent by each node are $P_1, P_2, \ldots, P_{J-1}$. Define $L = (43 - 35)/(J - 1) = 8/(J - 1)$, choose $r \in (0, 1)$, then we have

$$P_1 = Lr = L(-1)^{1-1}r^1 = Lr^1, \quad \text{positive}$$

$$P_2 = Lr(-r) = L(-1)^{2-1}r^2 = -Lr^2, \quad \text{negative}$$

$$P_3 = Lr(-r)(-r) = L(-1)^{3-1}r^3 = Lr^3, \quad \text{positive}$$

$$\vdots$$

$$P_{J-1} = L(-1)^{J-2}r^{J-1}.$$

The piece calculating method shows the following effects.

(i) The odd piece is positive and the even one emerges as a negative value.

(ii) If J is an odd number, the numbers of positive and negative piece are the same, that is, $(J - 1)/2$; otherwise, if J is an even number, there is one more positive piece.

(iii) We have $\Sigma_m \text{Positive Piece}_m \geq \Sigma_n |\text{Negative Piece}_n|$.

Now we define a sending time schedule based on technique of randomized time slot factor. Let a single time slot be $t_s = T_{\text{collusion}}/(J - 1)$, $T_{\text{collusion}}$ represents time spent in collusion phase. The forwarding moment is determined as.

t_1: a random moment between 0 and t_s;

t_2: a random moment between t_s and $2t_s$;

$\vdots$

t_{J-1}: a random moment between $(J-2)\,t_s$ and $(J-1)\,t_s$.

Then, only the node from node set T sends the piece P_i at the time t_i $(i = 1, 2, \ldots, J - 1)$ to neighbor.

Theorem 1. *Under the same collision rate p, piece loss in PEPDA is less than that in SMART.*

Proof. In SMART scheme, for the ith node, we have

$$\begin{aligned} P_1 &= D_i r_1, \text{ the rest is } D_i - D_i r_1 = D_i(1-r_1), \\ P_2 &= D_i(1-r_1)r_2, \text{ the rest is } D_i(1-r_1) - D_i(1-r_1)r_2 \\ &= D_i(1-r_1)(1-r_2), \\ &\vdots \\ P_{J-1} &= D_i r_{J-1} \prod_{j=1}^{J-2}(1-r_j), \end{aligned} \tag{3}$$

where $r_1, r_2, \ldots, r_{J-1} \in (0,1)$.

The sum of sent pieces at ith node is $\sum_{m=1}^{J-1} P_m$. If the rate of lost piece is p, then the loss is

$$p\sum_{m=1}^{J-1} P_m = pD_i \left[r_1 + (1-r_1)r_2 + (1-r_1)(1-r_2)r_3 + \cdots + r_{J-1}\prod_{j=1}^{J-2}(1-r_j) \right]. \tag{4}$$

The total loss R_{SMART} is

$$\sum_{i=1}^{N-1} pD_i \left[r_1 + (1-r_1)r_2 + (1-r_1)(1-r_2)r_3 + \cdots + r_{J-1}\prod_{j=1}^{J-2}(1-r_j) \right]. \tag{5}$$

Similarly, in PEPDA scheme, the sum of sent pieces at ith node is

$$\sum_{m=1}^{J-1} P_m = D_i\left[r_1 - r_1^2 + r_1^3 + \cdots + (-1)^{J-2} r_1^{J-1} \right]. \tag{6}$$

Then the loss is

$$pD_i\left[r_1 - r_1^2 + r_1^3 + \cdots + (-1)^{J-2} r_1^{J-1} \right]. \tag{7}$$

Therefore, the total loss R_{PEPDA} is

$$\sum_{i=1}^{N} pD_i\left[r_1 - r_1^2 + r_1^3 + \cdots + (-1)^{J-2} r_1^{J-1} \right]. \tag{8}$$

We need now to prove $R_{\text{SMART}} > R_{\text{PEPDA}}$. In fact, for $N = 1$, we have $R_{\text{SMART}} = R_{\text{PEPDA}}$.

For $N \geq 2$ and N is an odd number, we have

$$\begin{aligned} R_{\text{SMART}} - R_{\text{PEPDA}} &= (1-r_1) \times \left[r_2 + (1-r_2) \times r_3 + \cdots + r_n \times \prod_{j=2}^{n-1}(1-r_j) + r_1^2 + r_1^4 + \cdots + r_1^{n-1} \right] > 0. \end{aligned} \tag{9}$$

If N is an even number, we have

$$\begin{aligned} R_{\text{SMART}} - R_{\text{PEPDA}} &= (1-r_1) \times \left[r_2 + (1-r_2) \times r_3 + \cdots + r_n \times \prod_{j=2}^{n-1}(1-r_j) + r_1^2 + r_1^4 + \cdots + r_1^{n-2} \right] \\ &\quad + r_1^n > 0. \end{aligned} \tag{10}$$

This gives the conclusion. The theorem somehow indicates that under the same collision rate, aggregation accuracy in PEPDA is superior to that of SMART. Moreover, the collision rate in our proposed scheme is reduced; accordingly, the practical aggregation accuracy will be even better as a result of smaller piece loss. □

5.3. Postback Phase. This phase estimates the data loss rate for each node, which will be used in calculating the compensation factor.

During the collusion phase, an ACK forwarding table is established. After the collusion phase, each node sends ACK messages to nodes recorded in the table. We could do this step once a node is stored in the table. But this will cause more data collision. So the ACK message is sent after collusion phase to avoid collision between piece packet and ACK packet. Each node in the node set T will receive the ACK message from neighbor. Let B_i be the number of ACK messages received by the node i, then, the difference $(J-1)-B_i$ implies the number of packet lost in the collusion phase. Then, we have the rate of lost piece:

$$\tau_i = \frac{J-1-B_i}{J-1}. \tag{11}$$

According to this rate, the node i calculates the compensation factor as follows:

$$C_i = L\left(r - r^2\right)\frac{J-1-B_i}{J-1}. \tag{12}$$

5.4. Aggregation. Before aggregating, node i adds C_i to itself as compensation, and all nodes in the network then do in-network aggregation following TAG protocol proposed in [17].

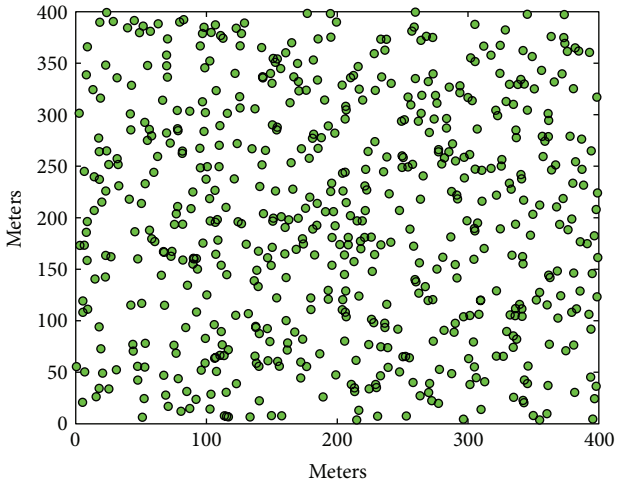


Figure 6: Nodes distribution.

For an individual node, the total piece value to be sent is

$$\begin{aligned} &P_1 + P_2 + \cdots + P_{J-1} \\ &\quad = Lr^1 - Lr^2 + \cdots + L(-1)^{J-2} r^{J-1} \\ &\quad = L\left(r - r^2 + r^3 + \cdots + (-1)^{J-2} r^{J-1}\right) \\ &\quad > L\left(r - r^2\right), \quad r \in (0,1). \end{aligned} \tag{13}$$

From the formulas (12) and (13), it indicates that the compensation C_i used in PEPDA algorithm is the minimum of lost pieces.

6. Simulation and Analysis

This section evaluates performances of PEPDA scheme in terms of privacy-preserving efficacy, piece accepting rate, aggregation accuracy, and power analysis. The simulation environment is TOSSIM under TinyOS. The parameters in simulation are shown in Table 1. A topology of nodes is shown in Figure 6.

6.1. Privacy-Preserving Efficacy. We denote by $P(q)$ the probability that private data is disclosed and take it as a privacy-preserving efficacy metric, where q represents the probability that link level privacy is broken.

In SMART scheme, the privacy is broken only when $J-1$ outgoing links and all the incoming links are cracked. Accordingly, $P(q)$ can be approximately defined as

$$P(q) = q^{J-1} \sum_{k=0}^{d_{\text{in\_max}}} p\,(\text{in-degree} = k)\, q^k, \tag{14}$$

where $p(\text{in-degree} = k)$ is defined as the probability that the in-degree of a node is k and $d_{\text{in\_max}}$ represents the maximum in-degree in a network.

According to (14), theoretical privacy-preserving efficacy is shown in Figure 7, which indicates that $P(q)$ decreases with the increase of slicing number J. The theoretical value cannot be reached due to the problem of SFN, which limits outgoing links, so that the incoming links get reduced as well. In fact, a practical definition of $P(q)$ should be calculated as

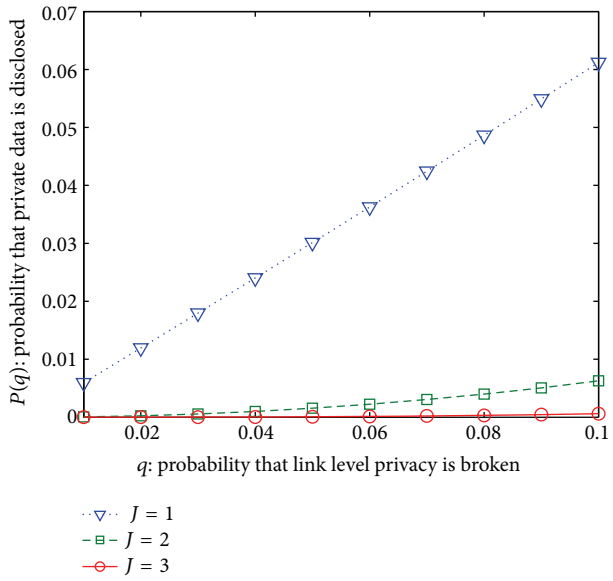


Figure 7: Theoretical value of $P(q)$ in SMART.

Table 1: Simulation parameters.

Radio parameters	White Gaussian noise	Noise floor
	4 dB	−105 dB
Topology parameters	Number of nodes	Terrain dimensions
	600	400 meters × 400 meters

$$P(q) = \sum_{k=0}^{d_{\max}} p\,(\text{degree} = k)\, q^k, \tag{15}$$

where $p(\text{degree} = k)$ represents the probability that the degree of a node is k, and $d_{\max}$ is the maximum degree in a network. Note that the degree of a node includes both in-degree and out-degree values.

Under this circumstance, $P(q)$ fluctuates according to slicing number J, which is shown in Figure 8. Although there is a tiny fluctuation, $P(q)$ keeps decreasing with the increase of slicing number J overall.

In PEPDA scheme, only if an eavesdropper breaks all the incoming and outgoing links, along with the end-to-end encryption key of a node, will it be able to crack the private data held by this node. Therefore, $P(q)$ can be approximately defined as

$$\begin{aligned} P(q) = P_e \Bigg(& P_T q^{J-1} \sum_{k=0}^{T_{\max}} p_T\,(\text{in-degree} = k)\, q^k \\ & + P_F \sum_{l=0}^{F_{\max}} p_F\,(\text{in-degree} = l)\, q^l \Bigg), \end{aligned} \tag{16}$$

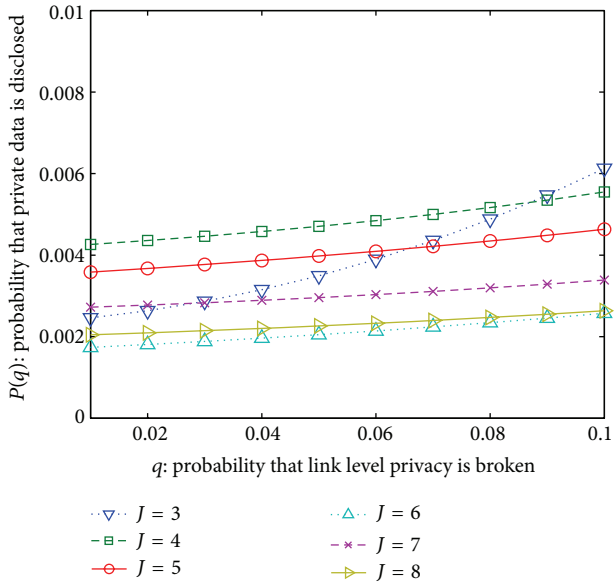


Figure 8: Practical value of $P(q)$ in SMART.

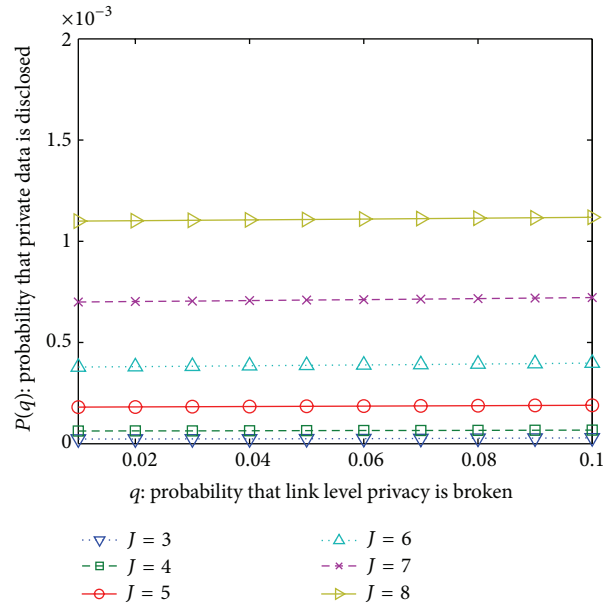


Figure 9: $P(q)$ in PEPDA.

where P_e is the probability that an end-to-end encryption key is cracked and can be approximately valued as $P_e = 1/N$, P_T and P_F are the percentages of the sets T and F, respectively, that is, $P_T = |T|/N$ and $P_F = |F|/N$. $T_{\max}$ and $F_{\max}$ are the maximums of the in-degree in the node sets T and F, respectively. $p_T(\text{in-degree} = k)$ is the probability that the in-degree of a node in the node set T is k, namely, $p_T(\text{in-degree} = k) = N(\text{in-degree} = k)/|T|$, where $N(\text{in-degree} = k)$ is the number of node which in-degree equals to k. We have the same definition for $p_F(\text{in-degree} = l) = N(\text{in-degree} = l)/|F|$.

Figure 9 shows the privacy-preserving efficacy $P(q)$ of PEPDA scheme. As is illustrated in Figure 9, $P(q)$ increase as J grows, which is different from that in SMART scheme. In the original SMART scheme, outgoing and incoming links of a node increases as J grows, making it tougher to crack private data. Therefore, privacy-preserving efficacy improves. Figures 8 and 9 imply that PEPDA demonstrates a better performance than SMART in terms of privacy preserving.

As mentioned in [14], applying homomorphism encryption technique in end-to-end encryption needs to tackle with several underlying issues, for example, capacity of confidentiality protection. Our scheme would not have to worry about this problem. If $D_i' = D_i + k_i$ and D_i' becomes D_i'' after collusion communication, D_i'' in node set T is no longer D_i', as these nodes have done slice and mix operations; D_i'' in node set F may not be D_i' either, as these nodes may receive piece from nodes in node set T. Therefore, even though the range of k_i is known by the adversary, it is still difficult to guess the range of D_i, which identifies the original private data in a node. In collusion phase, if a node belongs to node set F and receives piece from node set T, it is called an infected node. We define IR (infected rate) as $\text{IR} = |N_I|/|F|$, where N_I is the number of infected node. IR reveals what the set T influences node set F; the bigger the IR value is, the more difficult it is for adversary to extract the original private data in node set F. As is illustrated in Figure 10, more than half the nodes from node set F are infected when J is lower than 7. The curve reaches its peak when slicing number J is selected as 4. The set F contains more 550 nodes when $J = 8$, as shown in Figure 2. Only about 50 nodes in the set T slice data into J pieces.

6.2. Piece Accepting Rate. Because collision happens during the collusion phase, some data pieces are lost. It influences effectively aggregation accuracy. To reduce the collision rate is one of the objectives in designing data aggregation approach. As a metric, piece accepting rate is evaluated. It is defined by

$$p_r = \frac{\sum_{i=1}^{N} N_i^r}{\sum_{j=1}^{N} N_j^s}, \tag{17}$$

where N_i^r is the number of received pieces by a node i, while N_j^s is the number of pieces sent by a node j.

Figure 11(a) shows the relationship between slicing number J and piece accepting rate by using the partial factor. It indicates that with the increase of slicing number J, piece accepting rate increases. Particularly, piece accepting rate approaches 1 when $J = 11$. That is because, as described in Figure 2, the set F contains almost all nodes in this case, only a few nodes need to be sliced. Therefore, piece collision is reduced. Figure 11(b) describes the influence of piece accepting rate with the factors with $J = 3$. The curves of both SMART scheme and the partial factor schemes overlap. The reason is that the number of nodes in the set F is too small, say $|F| = 24$ for this case. Therefore, it has little influence on the piece collision by using the technique of the partial factor. However, there is a significant increase in piece accepting rate with randomized time slot factor, because the pieces sending schedules are optimized, then the collision rate is reduced. Obviously, compared with SMART scheme, Figure 11(b) shows a good performance of piece accepting

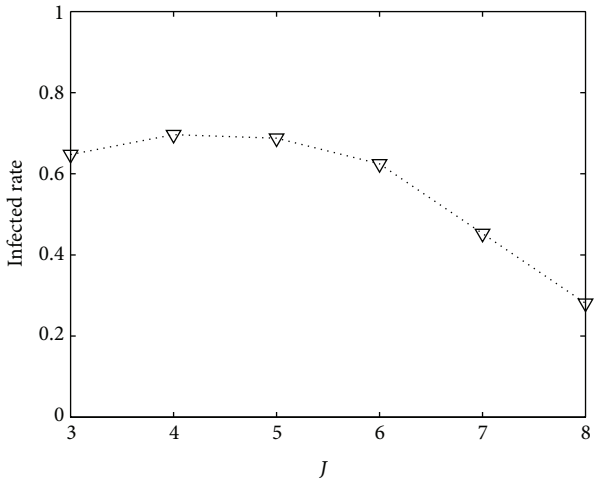
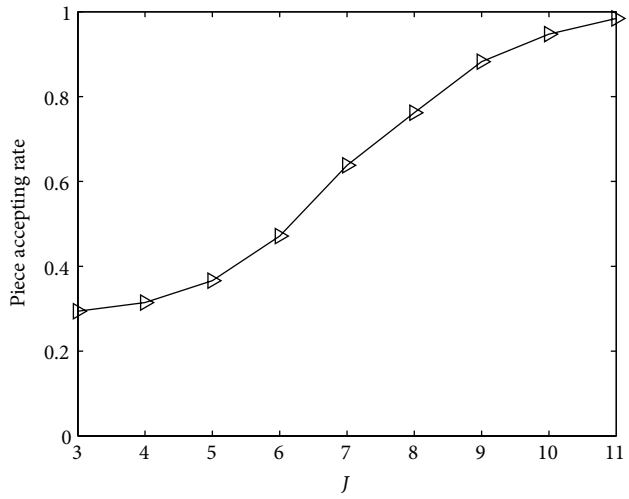
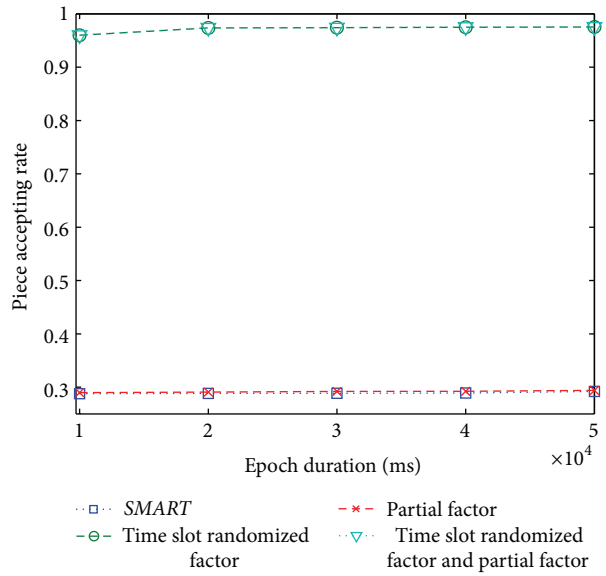


Figure 10: Infected rate.



(a) Partial Factor



(b) Piece accepting rate comparison

Figure 11: Piece accepting rate influenced by the factors.

rate by using the randomized time slot factor, which in turn enhances aggregation accuracy.

6.3. Aggregation Accuracy. Figure 12 shows the aggregation accuracy of PEPDA with respect to slicing number J. The accuracy curve rises as J increases at beginning and then keeps steady, though there is tiny fluctuation within, because we use the compensation factor and the small data factor. However, for SMART scheme, the aggregation accuracy decreases as J increases.

Figure 13 gives an accuracy comparison between SMART and PEPDA schemes with $J = 3$. Obviously, the accuracy of PEPDA scheme is twice as that of SMART scheme. This is a result by using five factors to optimize the aggregation algorithms with different sides. The randomized time slot factor and the partial factor are used to reduce the collision rate, while the small data factor and the positive and negative factor are applied to reduce the loss caused by the collision. The compensation factor is employed to correct the loss. Figure 12 demonstrates a good performance of PEPDA scheme with the five factors.

6.4. Complexity Analysis. This subsection focuses on evaluating complexity of schemes in terms of communication overhead and computation overhead.

6.4.1. Communication Overhead. Each node needs 2 basic messages in both SMART and PEPDA schemes. One is a *Hello* message to accomplish tree formation; another is for data aggregation [7]. Except for these common overheads, in PEPDA, an extra communication overhead consists of collusion overhead and ACK forwarding overhead. In fact, each node in the set T sends $(J-1)$ pieces in the collusion phase and B_p ACK messages in the postback phase. Then the total overhead of collusion communication is $(J-1)|T|+B_1+B_2+\cdots+B_{|T|}$, where B_i $(i = 1, 2, \ldots, |T|)$ are the piece accepting rates. As for SMART scheme, collusion communication is exactly the extra communication overhead, which is $(J-1)N$.

Figure 14 gives communication overheads of PEPDA. It shows that the total communication overhead is approximately twice as much as collusion communication overhead. Therefore, we have $(J-1)|T| \approx B_1 + B_2 + \cdots + B_{|T|}$, which implies that $B_i \approx (J-1)$ $(i = 1, 2, \ldots, |T|)$ in this case. This is because of the randomized time slot factor, which increases the piece accepting rate and makes it approach 1.

6.4.2. Computation Overhead. Both in SMART and PEPDA, a node executes the following process during collusion duration:

(i) calculate the piece size (slice) → encrypt → send

(ii) receive → decrypt → sum (mix).

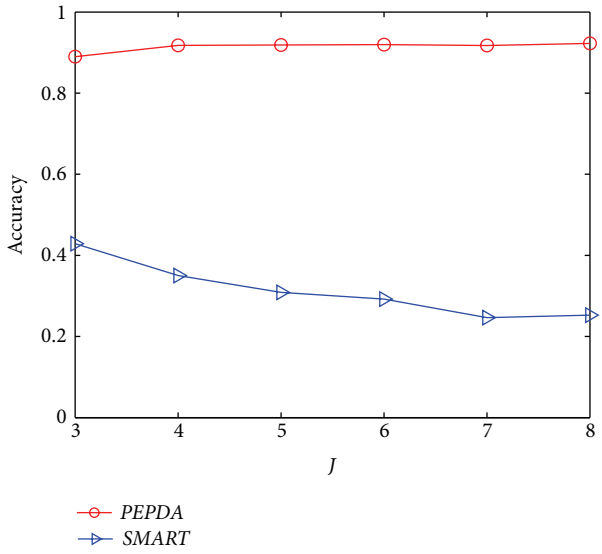


Figure 12: Accuracy P_a versus slicing number J.

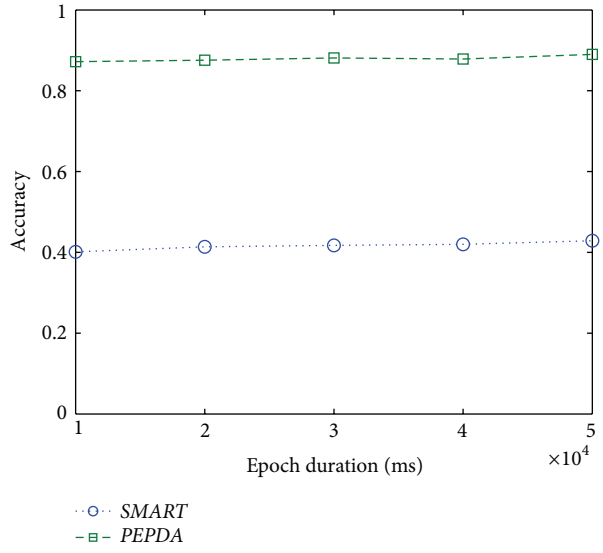


Figure 13: Accuracy comparison.

Therefore, the computational energy consumption can be determined by the following equation:

$$E_j = N_e\text{Enc} + N_d\text{Dec} + N_c\text{Cal}, \quad (18)$$

where Enc and Dec are the energy costs of doing one time encryption and decryption of 10 bits value, Cal represents the energy required to perform one time computational instructions, N_e, N_d, and N_c are the numbers of operations of encryption, decryption, and computation, respectively.

In SMART, all nodes perform $(J-1)N$ times operations in both encryption and decryption phases to achieve hop-by-hop encryption. Meanwhile, it takes $(J-1)N$ times operations to compute in both slicing and mixing phases; while in PEPDA, as a result of SFN, slice and mix computing operations, together with encrypt and decrypt operations, are

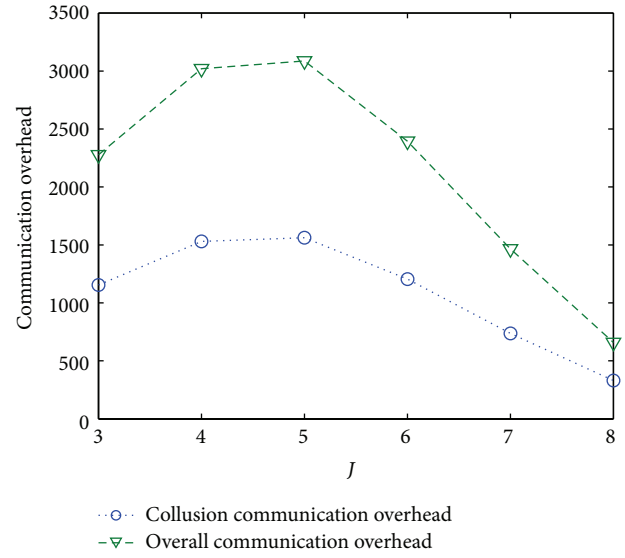


Figure 14: Communication overhead in PEPDA.

Table 2: Detailed values of parameters for (1) in SMART and PEPDA.

Scheme	N_e	N_d	N_c
SMART	$(J-1)N$	$(J-1)N$	$2(J-1)N$
PEPDA	$(J-1)\lvert T\rvert$	$(J-1)\lvert T\rvert$	$2(J-1)\lvert T\rvert$

Table 3: Energy consumption of common operations on MICAz Mote and TelosB Mote [9].

Operations	MICAz	TelosB
Compute for 1 clock tick	3.5 nJ	1.2 nJ
Transmit 1 bit	0.6 μJ	0.72 μJ
Receive 1 bit	0.67 μJ	0.81 μJ

declining to $(J-1)|T|$ (see Table 2). In order to get more precise result, we take MICAz and TelosB as examples [9]. The MICAz has a bus width of 8 bits and runs at 7.37 MHz, and the TelosB features the 16-bit microcontroller running at 4 MHz.

The energy consumption of encrypting and decrypting 10 bits of data on the MICAz and TelosB architectures with IDEA, RC4, and RC5 algorithms for hop-by-hop encryption can be found in [9] (Table 3). Figure 15 illustrates the total energy cost of each scheme with the change of slicing number J. Total energy cost of PEPDA is lower than SMART.

Table 4 summarizes properties of SMART, ESPART, and PEPDA in terms of security, data accepting rate, aggregation accuracy, and overhead. The PEPDA scheme demonstrates a good performance compared with the other two methods.

7. Conclusion

Accuracy and privacy preserving are two important challenges in designing data aggregation algorithm in WSNs. Based on SMART scheme, the five factors are used to optimize the algorithms of data aggregation. The objective is

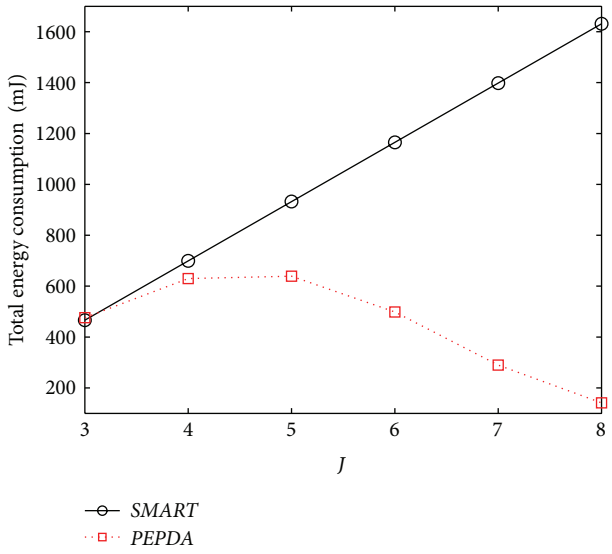


Figure 15: Total energy cost of SMART and PEPDA.

Table 4: Performance comparison.

Evaluation	SMART	ESPART [8]	PEPDA
Privacy-preserving efficacy	Excellent	Excellent	Excellent
Piece accepting rate	Low	Ideal	High
Aggregation accuracy	Low	Ideal	High
Communication overhead	Large	Fair	Small
Computation overhead	Fair	Small	Small

to reduce the collision rate and collision loss. From this point, the randomized time slot factor and the partial factor are developed to decrease the collision rate, while the small data factor, the positive and negative factor, and the compensation factor are designed to improve the collision loss. We propose a novel privacy-preserving data aggregation scheme based on the five optimized factors. Analysis and simulation show that the proposed PEPDA scheme demonstrates a good performance in terms of accuracy, complexity, and security.

From the point of view of security, PEPDA uses the same mechanism as that in SMART. It is interesting to design privacy-preserving data aggregation schemes by combining hop-by-hop encryption with end-to-end encryption in PEPDA.

Acknowledgments

This work is supported by the National Basic Research Program (973 Program) of China under Grant no. 2011CB302903, the National Natural Science Foundation of China under Grants nos. 61272084, 61202004, and 61202353, the Key Project of Natural Science Research of Jiangsu University under Grant no. 11KJA520002, and Specialized Research Fund for the Doctoral Program of Higher Education under Grant no. 20113223110003.

References

[1] H. Chan, A. Perrig, and D. Song, *Secure Hierarchical In-Network Aggregation in Sensor Networks*, ACM CCS, Alexandria, Va, USA, 2006.

[2] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: a secure hop-by-hop data aggregation protocol for sensor networks," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '06)*, pp. 356–367, Florence, Italy, May 2006.

[3] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of ACM Conference on Computer and Communications Security (CCS '02)*, pp. 41–47, Washington, DC, USA, November 2002.

[4] H. Chan, A. Perrig, and D. X. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of Symposium on Security and Privacy (SP '03)*, pp. 197–213, Carnegie Mellon Univ., Pa, USA, May 2003.

[5] D. G. Liu and N. Peng, "Establishing pairwise keys in distributed sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 1, pp. 41–77, 2005.

[6] S. Ruj, A. Nayak, and I. Stojmenovic, "Fully secure pairwise and triple key distribution in wireless sensor networks using combinatorial designs," in *Proceedings of the 30th IEEE International Conference on Computer Communications (IEEE INFOCOM '11)*, pp. 326–330, Shanghai, China, April 2011.

[7] W. B. He, X. Liu, and H. Nguyen, "PDA: privacy-preserving data aggregation in wireless sensor networks," in *Proceedings of the 26th Annual IEEE Conference on Computer Communications (IEEE INFOCOM '07)*, pp. 2045–2053, Anchorage, Alaska, USA, May 2007.

[8] G. Yang, A. Q. Wang, and Z. Y. Chen, "An energy-saving privacy-preserving data aggregation algorithm," *Chinese Journal of Computers*, vol. 34, pp. 792–800, 2011.

[9] M. M. Groat, W. Hey, and S. Forrest, "KIPDA: K-indistinguishable privacy-preserving data aggregation in wireless sensor networks," in *Proceedings of the 30th IEEE International Conference on Computer Communications (IEEE INFOCOM '11)*, pp. 2024–2032, April 2011.

[10] S. Ozdemir and Y. Xiao, "Secure data aggregation in wireless sensor networks: a comprehensive overview," *Computer Networks*, vol. 53, no. 12, pp. 2022–2037, 2009.

[11] R. Bista, D. Kim, and J. Chang, "A new private data aggregation scheme for wireless sensor networks," in *Proceedings of Computer and Information Technology (CIT '10)*, pp. 273–280, West Yorkshire, UK, June 2010.

[12] C. Castelluccia, E. Mykletun, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems—Networking and Services (MobiQuitous '05)*, pp. 109–117, San Diego, Calif, USA, July 2005.

[13] E. de Cristofaro, "A secure and privacy-protecting aggregation scheme for sensor networks," in *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM '07)*, pp. 1–5, Espoo, Finland, June 2007.

[14] T. M. Feng, C. Wang, W. S. Zhang, and L. Ruan, "Confidentiality protection for distributed sensor data aggregation," in *Proceedings of the 27th Conference on Computer Communications (INFOCOM '08)*, pp. 56–60, Phoenix, Ariz, USA, April 2008.

[15] W. S. Zhang, C. Wang, and T. M. Feng, "GP^2S: generic privacy-preservation solutions for approximate aggregation of sensor

data, concise contribution," in *Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '08)*, pp. 179–184, Hong Kong, March 2008.

[16] A. F. Chan, "A security framework for privacy-preserving data aggregation in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 7, no. 4, pp. 29–45, 2011.

[17] S. Madden, M. J. Franklin, and J. M. Hellerstein, "TAG: a tiny aggregation service for ad-hoc sensor networks," in *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI '02)*, pp. 131–146, New York, NY, USA, 2002.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 791814, 9 pages
http://dx.doi.org/10.1155/2013/791814

Research Article

Perceptual Hashing-Based Robust Image Authentication Scheme for Wireless Multimedia Sensor Networks

Hongxia Wang and Bangxu Yin

School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China

Correspondence should be addressed to Hongxia Wang; hxwang@home.swjtu.edu.cn

Received 29 January 2013; Accepted 13 March 2013

Academic Editor: Muhammad Khurram Khan

Image authentication is critical for secure image transmission and storage in a wireless multimedia sensor network (WMSN). In this paper, we propose a perceptual hashing-based robust image authentication scheme, which applies the distributed processing strategy for perceptual image hashes and can provide compactness, visual fragility, perceptual robustness, and security in digital image authentication for WMSN. In the proposed scheme, first, the cluster head node generates a secure pseudorandom chaotic sequence with keys and sends it to the image capturing node, then the image capturing node uses the received chaotic sequence to divide randomly the captured image into several overlapping rectangles, after that, two gravity centers of each rectangular region block are calculated, and finally the binary distance of the two gravity centers will be calculated in each general cluster member node. The cluster head node receives the binary sequence of the distance from all of the general cluster member nodes and combines them to form the perceptual hashing sequence to be sent to the base station for image authentication purpose. Experimental results show that the proposed scheme has satisfactory authentication ability and can ensure not only the visual fragility for perceptually distinct images but also robustness for perceptually identical images via image rotation, JPEG compression, and noise burring.

1. Introduction

Wireless sensor networks (WSNs) are going to be widely used in the near future due to their breadth of applications by military, exploration teams, researchers, and so on. Most of this research is concerned with scalar sensor networks that measure physical phenomena, such as temperature, light, humidity, pressure, acoustic sensor, or location of objects that can be conveyed through low-bandwidth and delay-tolerant data streams. Recently, the focus is shifting toward research aimed at revisiting the sensor network paradigm to enable delivery of multimedia information, such as a monitoring data, digital image, audio and video streams, as well as scalar data [1]. This effort will result in distributed, networked systems, referred to in this paper as wireless multimedia sensor networks (WMSNs). The WMSNs will enable new applications such as multimedia surveillance, traffic enforcement and control systems, advanced health care delivery, structural health monitoring, and industrial process control. Consequently, it will bring new security of challenges as well as new opportunities. Secure and robust multimedia communications become increasingly important for energy-constrained WMSNs [2]. As one of the security techniques, image authentication is critical for secure image transmission and storage in WMSNs. However, conventional data authentication schemes cannot be applied directly to WMSN due to the constraints on limited energy and computing resources in sensor nodes. Those constraints pose great challenges to WMSN development and motivate us to design a proper authentication strategy for WMSNs.

For open communication channel, WSNs are vulnerable to various attacks, and the security in WSNs is required. A secure hierarchical key management scheme in WSNs was proposed in [3]. The security analysis and simulation show the scheme can prevent several attacks effectively and reduce the energy consumption. In WSNs communications, Han et al. [4] described six types of attacks including communication attacks, attacks against privacy, sensor node targeted attacks, power consumption attacks, policy attacks, and cryptology attacks on key management. In communication attacks, eavesdropper can easily access or even manipulate message such as injecting, cropping, and tampering. So the receiver

needs to make sure that the data used in any decision-making process originates from the correct source. Data authentication prevents unauthorized parties from participating in the networks and legitimate nodes should be able to detect messages from authorized nodes and reject them. In [5], a robust user authentication scheme for WSNs was proposed. The scheme takes an advantage of the two-factor authentication concept to provide a secure authentication system offering balanced features in terms of security and performance.

The authenticity of data and commands is also a critical requirement for the correct behavior of a WMSN [6]. Digital images are becoming widely used in WMSNs as a kind of common multimedia information. Therefore, the key establishment technique should guarantee that the image communication and storage have a way for verifying the authenticity, creditability, and integrity of the received image in WMSNs. However, resource constraints in sensor networks (such as limited battery power and bandwidth/computation capability) pose challenges for the image authentication technique. Conventional binary data authentication schemes could provide data integrity in a strict sense regardless of multimedia content. However, those schemes are not applicable to WMSNs because only simple bit errors during data transmission can lead to the authentication failing in spite of preserving multimedia content [7, 8]. On the other hand, watermarking-based image content authentication techniques are robust against bit errors, packet losses, and compression distortion. However, watermark embedding creates extra source coding overheads and complicates transmission protocol design in WMSNs [9], which don't adapt well to WMSNs due to the constraints on limited energy and computing resources.

In [10], an optimized content-aware authentication scheme for JPEG 2000 images over lossy channels was proposed. An acyclic authentication graph was developed to optimize the trade-off between the expected image distortion and the cryptohash tagging cost, through the computation based on packet loss probability and visual importance level of the image packets. The work reported in [11] proposed a JPEG 2000 compatible stream authentication scheme that significantly reduced the computational complexity and had only a minimal authentication dependency overhead in WMSNs. Moreover, an authentication-aware wireless network resource allocation scheme was developed to reduce image distortion and energy consumption during transmission. The scheme significantly improved the authenticated image quality even under strict communication energy consumption constraints in WMSN. In [12], a rate-distortion optimization authentication scheme for H.264 video transmissions was proposed. A video packet transmission scheduler was designed to minimize the visual distortion under the limitation of total bit budget and authentication dependency. Another related work regarding bite rrors robust image or video authentication was given in [13–15]. However, all of these approaches are not able to be applied directly to WMSNs due to the energy constraint. In this paper, we propose a perceptual hashing-based robust authentication scheme for WMSNs. Based on the distributed processing strategy for perceptual image hashes, the proposed scheme can provide compactness, visual fragility, perceptual robustness, and security for image authentication in WMSNs.

2. Perceptual Image Hashing Extraction

A perceptual image hashing function maps an image to a short binary string as a digest based on an image's appearance to the human eye. Perceptual image hashing is a class of one-way mappings from image presentations to a perceptual hash value in terms of their perceptual content. Given an image I and its perceptually similar copy with minor distortion I_d, the image hashing function $H_k(\cdot)$ depends on the secret key k. In [16], the desirable properties of perceptual hashing function $H_k(\cdot)$ can be summarized as follows.

(i) *One-Way Function*. Ideally, the hash generation should be noninvertible:

$$I \longmapsto H_k(I). \tag{1}$$

(ii) *Compactness*. The size of the perceptual hashing value should be much smaller than that of the original image I

$$\text{Size}\left(H_k(I)\right) \ll \text{Size}(I). \tag{2}$$

(iii) *Perceptual Robustness*. Perceptually identical images should have similar perceptual hashing values

$$\Pr\left\{H_k(I) \approx H_k\left(I_d\right)\right\} \geq 1-\varepsilon, \quad 0 \leq \varepsilon < 1. \tag{3}$$

(iv) *Visual Fragility*. Perceptually distinct images should have different perceptual hashing values

$$\Pr\left\{H_k(I) \neq H_k\left(I'\right)\right\} \geq 1-\tau, \quad 0 \leq \tau < 1. \tag{4}$$

(v) *Security*. The perceptual hashing is intractable without the secret key

$$\Pr\left\{H_k(I) \neq H_{k'}(I)\right\} \geq 1-\delta, \quad 0 \leq \delta < 1. \tag{5}$$

All of the above parameters ε, τ, and δ should be close to zero.

Based on THE above properties, perceptual image hashing can be usually applied to image content identification, image indexing, content authentication, and so forth. In particular, a perceptual hash function should have a property, that is, two images that look the same map to the same hash value, even if the images have small bit-level differences. This differentiates a perceptual hash from traditional cryptographic hashes, such as SHA-l and MD5. In cryptography, the hash function is typically used for digital signature to authenticate the message being sent so that the receiver can verify its source. A key feature of conventional hashing algorithms such as SHA-l and MD5 is that they are extremely sensitive to the input data; that is, changing even one bit of the input message will change the output dramatically. However, image data often undergoes various content-preserving manipulations such as lossy compression, channel additive

noise, image enhancement, scaling, bit errors and packet losses during wireless transmission and storage in WMSNs. These distortions are usually insignificant, and image hashes should be robust to unmalicious distortions. On the contrary, some malicious manipulations could introduce perceptually significant distortions, for example, object insertion, removal, and substitution, and it is desirable that the image hash is sensitive to perceptually significant attacks. Therefore, image hashes should be robust to unmalicious distortions, but sensitive to malicious manipulations for the image authentication purpose [17].

2.1. Image Random Blocking by Chaotic Sequence. In order to enhance the security of the perceptual hashing algorithm, we use a secure pseudorandom sequence with the key to divide randomly the digital image into C overlapping rectangles, and the key controls the number of rectangles and the pseudorandom sequence. The image blocking can also make up the disadvantage that the extracted image features can only describe global characteristics of an image.

As a phenomenon found in a nonlinear dynamic system, chaos is deterministic and random-like. Based upon the sensitive dependence of chaotic systems on their initial conditions, a large number of nonperiodic, continuous broadband frequency spectrum, noise-like, yet deterministic, and reproducible signals can be generated. So chaos is very useful for generating secure pseudorandom sequences.

The chaotic maps (6) and (7) are given by

$$Z_{n+1} = uZ_n\left(1 - Z_n\right), \quad n = 1, 2, \ldots, \tag{6}$$

$$S_{n+1} = \left(1 + 0.3\left(S_{n-1} - 1.08\right) + 379S_n^2 + 1001 \times Z_n^2\right) \bmod 3, \tag{7}$$

where $3.57 < u \leq 4$ is the chaotic system parameter and $0 < Z_1 < 1$ and $-1.5 < S_0$, $S_1 < 1.5$ are the initial values of the two chaotic systems. Z_n in formula (7) is generated by (6). When $u = 3.9$, we compute the value space of Z_1 as follows:

Suppose $Z_1 = \{0 < Z_1(i) < 1 \mid i = 1, 2, \ldots, L_1\}$ and L_1 is an integer which is large enough and generates chaotic sequences $Z = \{Z(i, j) \mid i = 1, 2, \ldots, L_1, j = 1, 2, \ldots, L_2\}$, where L_1 represents the number of chaotic sequences and L_2 means the length of each chaotic sequence. When $Z_1' = \{0 < Z_1(i) + d < 1 \mid i = 1, 2, \ldots, L_1\}$ and d is close to zero, another group of chaotic sequence $Z' = \{Z'(i, j) \mid i = 1, 2, \ldots, L_1, j = 1, 2, \ldots, L_2\}$ is generated. We use function $y = \Omega(d)$ to test the value space of Z_1 as follows:

$$y = \Omega(d) = \frac{\left[\sum_{i=1}^{L_1}\sum_{j=1}^{L_2}\left|Z(i, j) - Z'(i, j)\right|\right]}{\left(L_1 \times L_2\right)}. \tag{8}$$

The curve of function $y = \Omega(d)$ is shown in Figure 1. From Figure 1, it is seen that, when $d = 10^{-19}$, $y \approx 0$. So the value space of Z_1 is 10^{19}. Similarly, the value spaces of u, S_0, and S_1 are shown in Figures 1 and 2. They are 1×10^{15}, 3×10^{14}, and 3×10^{16}, respectively.

Therefore, when the difference of initial values or chaotic parameters of chaotic system is greater than some specific value, two different chaotic sequences will be generated. For Z_1, the difference should be greater than 10^{19} and similarly for others. Let u in (6) and S_0 as secret keys be denoted by k_1 and k_2, respectively. Consequently, we will use the generated chaotic sequences with keys k_1 and k_2 to divide randomly the digital image into C overlapping rectangles for security purpose.

According to the image size, we adaptively select proper bits of the chaotic sequence as the coordinate of x-axis and y-axis, length and width of the random region to prevent out boundary, denoted by a quaternion C (C_x, C_y, *length*, *width*). So the image is divided into C overlapping rectangular regions shown in Figure 3. Because the quaternion is randomly generated, the rectangular areas are random.

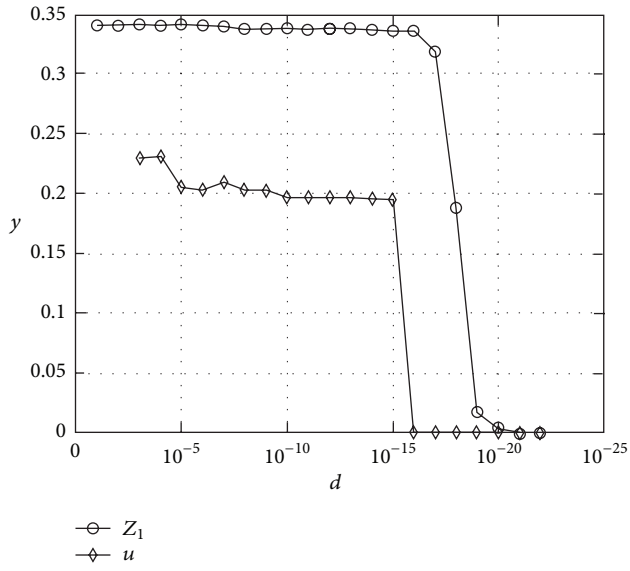


Figure 1: The value spaces of Z_1 and u.

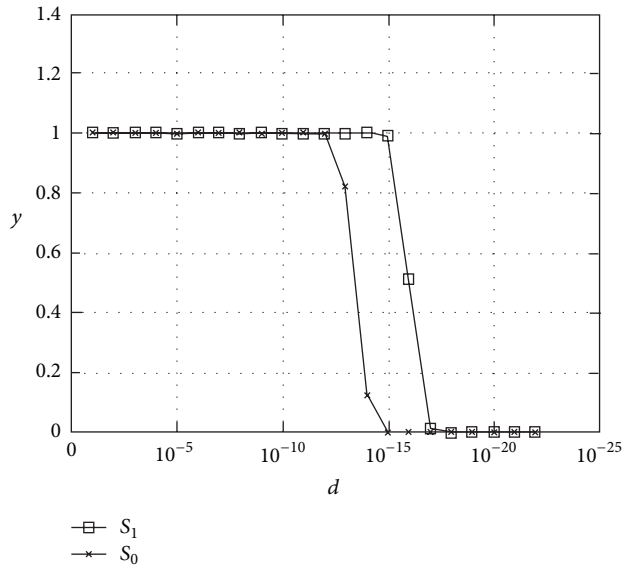


Figure 2: Value spaces of S_1 and S_0.

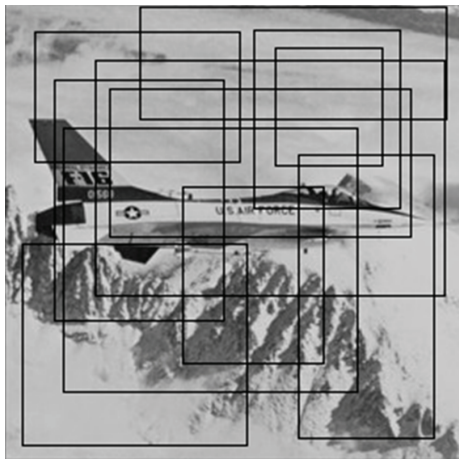


Figure 3: Image divided randomly into overlapping blocks.

2.2. Robust Local Feature Points Based on Gravity Center of Random Blocks. The local features of the image should be not only stable under geometric transforms such as rotation and scaling but also robust to insignificant distortions such as additive noising and blurring, bit errors, and packet losses during transmission in WMSNs. Based on the image random blocking in Section 2.1, we propose a robust local feature points extraction method using the gravity center of random blocks. The extracted robust local features will then be used to generate perceptual hashes in Section 2.3.

The two-dimensional (2D) moment can be directly used in the interested regions and does not need to separate them from the whole image. High-order moments are more sensitive to noise, while the low-order moments are insensitive to noise and bit errors, which is beneficial to the characterization of collectivity for the regions.

The 2D $(p + q)$th order origin moment of a continuous image I is defined as [18]

$$M_{pq} = \int_I x^p y^q f(x, y)\, dx\, dy, \qquad p, q = 0, 1, 2, \ldots, \tag{9}$$

where $f(x, y)$ represents the gray-level value at location (x, y). For digital image the integrals are replaced by summations and M_{pq} becomes

$$M_{pq} = \sum_{(x,y)\in I} x^p y^q f(x, y), \quad p, q = 0, 1, 2, \ldots. \tag{10}$$

The gravity center $G(m_x, m_y)$ of the image is defined in terms of the zero-order moment and first-order moments as follows:

$$m_x = \frac{M_{10}}{M_{00}}, \qquad m_y = \frac{M_{01}}{M_{00}}, \tag{11}$$

where the zero-order moment M_{00} represents the area of the image clearly.

In order to obtain the local feature, the coordinate of the gravity center $G_k(m_x^k, m_y^k)$ of each random rectangular block is calculated as

$$\begin{aligned} m_x^k &= \frac{\sum_x \sum_y x \cdot f^k(x, y)}{\sum_x \sum_y f^k(x, y)}, \\ m_y^k &= \frac{\sum_x \sum_y y \cdot f^k(x, y)}{\sum_x \sum_y f^k(x, y)}, \\ k &= 1, 2, \ldots, C, \end{aligned} \tag{12}$$

where $f^k(x, y)$ represents the gray-level value at location (x, y) in the kth random rectangular block and C is the number of random rectangular blocks. Thus, there are total C gravity centers of the pseudorandom rectangular blocks denoted by $G = \{G_1, G_2, \ldots, G_k, \ldots, G_C\}$. The local feature information of the image can be obtained by calculation of each block's gravity center, which improves the ability to distinguish different images.

The gravity center of the image has geometrically invariant property. In this paper, the rotation invariability is analyzed as an example. After a rotation by an angle θ about the original, the first-order moments are given by

$$\begin{aligned} M_{10}^r &= \sum_{(x,y)\in\Omega} (x\cos\theta + y\sin\theta) f(x, y) \\ &= M_{10}\cos\theta + M_{01}\sin\theta, \\ M_{01}^r &= \sum_{(x,y)\in\Omega} (y\cos\theta - x\sin\theta) f(x, y) \\ &= M_{01}\cos\theta - M_{10}\sin\theta \end{aligned} \tag{13}$$

and the zero-order moment $M_{00}^r = M_{00}$. So after a rotation by an angle θ, the changed gravity center is

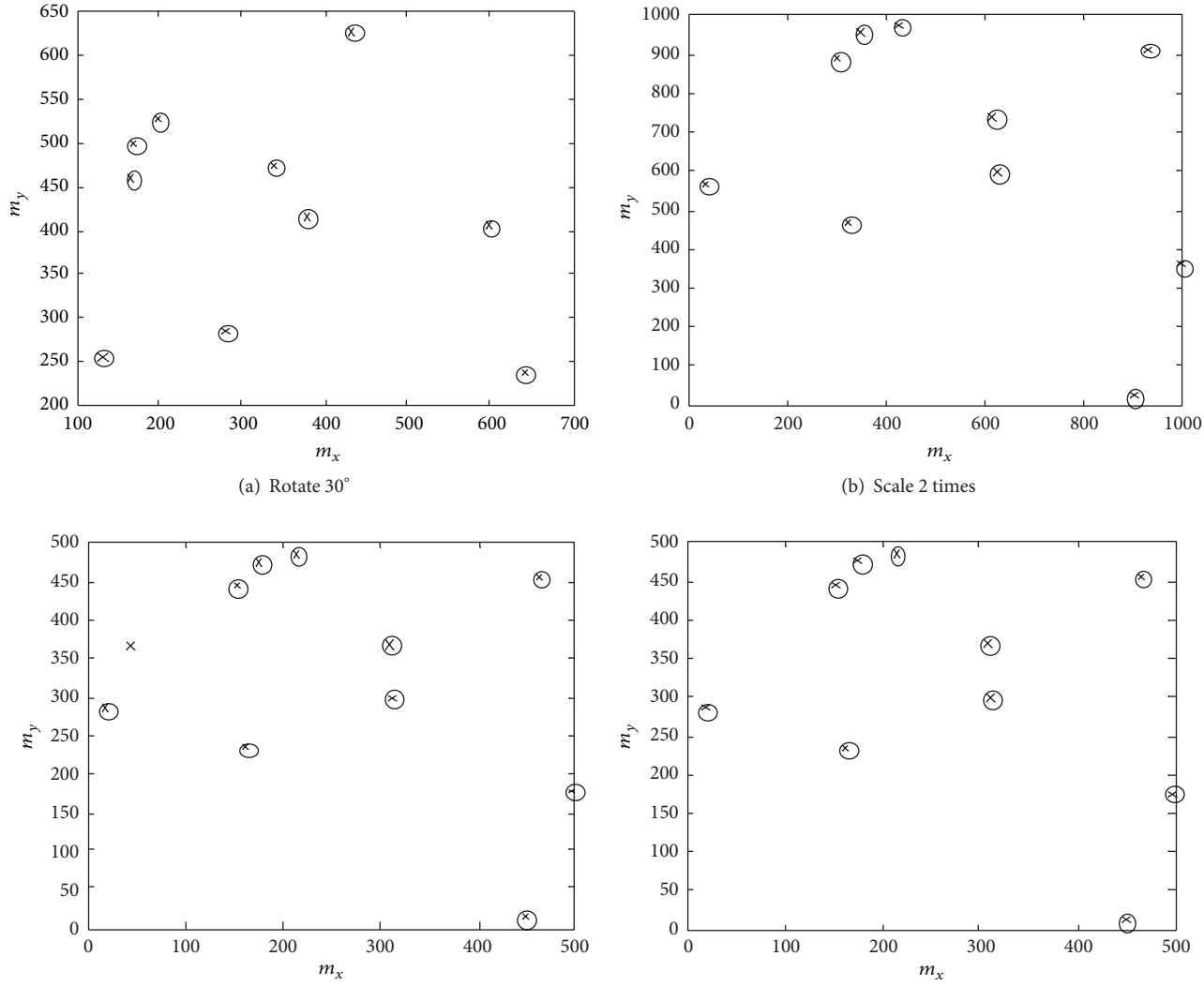
$$\begin{aligned} G_x^r &= \frac{M_{10}^r}{M_{00}^r} = G_x\cos\theta + G_y\sin\theta, \\ G_y^r &= \frac{M_{01}^r}{M_{00}^r} = G_y\cos\theta - G_x\sin\theta. \end{aligned} \tag{14}$$

Namely,

$$\begin{bmatrix} G_x^r \\ G_y^r \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} G_x \\ G_y \end{bmatrix}. \tag{15}$$

Thus, the rotation invariability is held. In similar analysis, other geometrical invariability characteristics can also be obtained.

Figure 4 shows the robustness of gravity centers under geometric distortions and common image processing. Note that × denotes the virtual locations gravity centers in the distorted image and o denotes theoretical locations of gravity centers. Once the two symbols are coincident, the geometrical invariability of gravity centers and the strong robustness to additive noise and JPEG compression will be indicated.



(a) Rotate 30°

(b) Scale 2 times

(c) Additive white Gaussian noise (variance is 46)

(d) JPEG compression (quality factor is 50%)

Figure 4: Robustness of gravity centers under geometric distortions and common image processing.

2.3. Perceptual Image Hashes Generation. The supplement image block of each random rectangular block is defined as

$$\widetilde{f}(x, y) = R_{\text{level}} - f(x, y), \tag{16}$$

where R_{level} is the maximum gray-scale level of the image. Likewise, we obtain the gravity center of the supplement image block. For each rectangular block, we calculate its supplement image block's gravity center and obtain total C gravity centers of the supplement image blocks. The normal gravity center of image usually lies around the center of image, so does that of the supplementary image. Thus the distance between the two gravity centers of image and its supplement image is short. We devise a solution by making a modification of the gravity center. The improved supplement image blocks' gravity center $\widehat{G}^k(\widehat{m}_x, \widehat{m}_y)$ of kth rectangular image block is obtained by

$$\begin{aligned} \widehat{m}_x^k &= \frac{\sum_x \sum_y x \cdot \exp\left(f^k(x, y)/\Delta_1\right)}{\sum_x \sum_y \exp\left(f^k(x, y)/\Delta_1\right)}, \\ \widehat{m}_y^k &= \frac{\sum_x \sum_y y \cdot \exp\left(f^k(x, y)/\Delta_1\right)}{\sum_x \sum_y \exp\left(f^k(x, y)/\Delta_1\right)}, \end{aligned} \tag{17}$$

where $\Delta_1 > 0$ is the quantification step. Through such modification, we enlarge the distance between the two gravity centers on one hand. On the other hand, the parameter Δ_1 guarantees the robustness against malicious attacks in calculating the two gravity centers.

In order to generate the perceptual image hashes, we calculate the distance of the two gravity centers between each rectangular block and its supplement image block. Considering the constraints on limited energy and computing

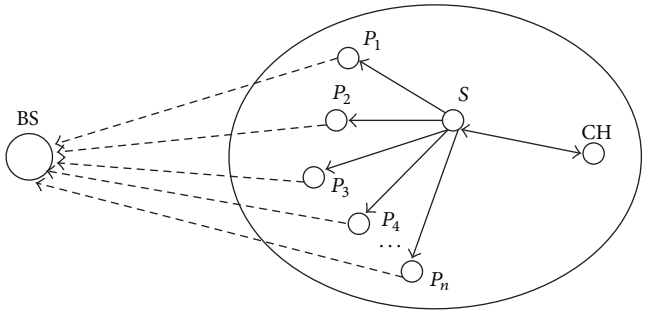


Figure 5: Structure of clustering in WMSNs.

resources in WMSNs, we use L_∞ norm to measure the spatial distance D_k between the locations of the two gravity centers for each rectangular block as follows:

$$D_k\left(G_k, \widehat{G}_k\right) = \max\left\{\left|m_x^k - \widehat{m}_x^k\right|, \left|m_y^k - \widehat{m}_y^k\right|\right\}, \quad k = 1, 2, \ldots, C. \tag{18}$$

Let $\Delta_2 > 0$ be another quantification step, and the distance D_k between the two gravity centers can be quantified as

$$\overline{D}_k = \left\lfloor \frac{D_k}{\Delta_2} \right\rfloor \Delta_2, \tag{19}$$

where $\lfloor \cdot \rfloor$ denotes the floor function. Obviously, the quantified distance $\overline{D}_k$ will be decimal integer. Moreover, the distance $\overline{D}_k$ is more robust against common image processing and the bit errors during transmission.

Then, we convert the decimal $\overline{D}_k$ to binary sequence $(b_{k,1}, b_{k,2}, \ldots, b_{k,j})_2$, $b_{k,j} \in \{0, 1\}$, and finally combine the binary sequences of each distance together to form the perceptual hashing sequence H as follows:

$$H = \left(b_{1,1}b_{1,2}\cdots b_{1,j}\right)_2 \left\|\left(b_{2,1}b_{2,2}\cdots b_{1,j}\right)_2\right\| \cdots \left\|\left(b_{k,1}b_{k,2}\cdots b_{k,j}\right)_2\right\| \cdots \left\|\left(b_{C,1}b_{C,2}\cdots b_{C,j}\right)_2\right. . \tag{20}$$

3. Distributed Processing Strategy for Perceptual Image Hashes

In WSNs, clustering expedites many desirable functions and provides many advantages such as load balancing, energy savings, and distributed key management. The most prominent benefit of clustering is that it can greatly reduce the energy consumption of nodes and lengthen the network lifetime [19]. In this paper, in order to adapt well to the limited power resources and computational capabilities in sensor nodes, we consider clustering-based WMSNs with densely distributed nodes. The structure of a clustering is shown in Figure 5. Each clustering consists of a cluster head (CH), several general cluster member nodes, and a camera sensor that captures the digital image. In Figure 5, *BS* represents the base station, CH is a cluster head node, *S* is the image capturing node, and $P_1 \sim P_n$ are the general cluster member nodes whose each node is assigned a fixed ID.

The distributed processing strategy is as follows:

Step 1. The cluster head node CH generates a secure pseudo-random chaotic sequence with the keys k_1 and k_2 according to the method in Section 2.1 and sends this chaotic sequence to the image capturing node *S*. In addition, the ID of each general cluster member node P_i $(i = 1, 2, \ldots, n)$ is also sent to node *S*.

Step 2. When the image capturing node *S* captures an image, it will use the received chaotic sequence to divide randomly the captured image into *C* overlapping rectangles. Moreover, each rectangle block is sent to the general cluster member node P_i. Note that the chaotic sequence is mapping to the ID of the general cluster member nodes one by one.

Step 3. The distance of the two gravity centers for each rectangle block will be calculated in each general cluster member node P_i, and its corresponding binary sequence $(b_{k,1}, b_{k,2}, \ldots, b_{k,j})_2$ can be generated by the method in Section 2.3, which is sent to the cluster head node CH.

Step 4. The cluster head node CH receives the binary sequences $(b_{k,1}, b_{k,2}, \ldots, b_{k,j})_2$, $k = 1, 2, \ldots, C$, from all of the general cluster member nodes, then combines them to form the perceptual hashing sequence, and finally the cluster head node CH sends the perceptual hashing sequence to the base station for image authentication purpose.

4. Perceptual Hashing-Based Image Authentication

When we identify the received image, firstly, the perceptual hashing sequence is obtained from the base station and is matched with the perceptual hashing sequence generated by the dubitable image. If the Hamming distance between two perceptual hashing sequences is less than the specified threshold, the image will be deemed an authentic version. Otherwise, the image is forged.

The image authentication framework is shown in Figure 6. The steps are as follows.

Step 1. The perceptual hashing sequence H_1 is received from the base station.

Step 2. The dubitable image is partitioned into *C* random rectangular blocks by the same secret keys k_1 and k_2 like perceptual hashing generation process described in Section 2.1; then the distance of the two gravity centers for each rectangle block is calculated and quantified; after that the robust feature is extracted. Thus another perceptual hashing sequence H_2 can be restructured like the description in Section 2.3.

Step 3. Setting a threshold $T > 0$, normalized Hamming distance is calculated by

$$DH\left(H_1, H_2\right) = \frac{1}{L}\sum_{k=1}^{L}\left|H_1(k) - H_2(k)\right|, \tag{21}$$

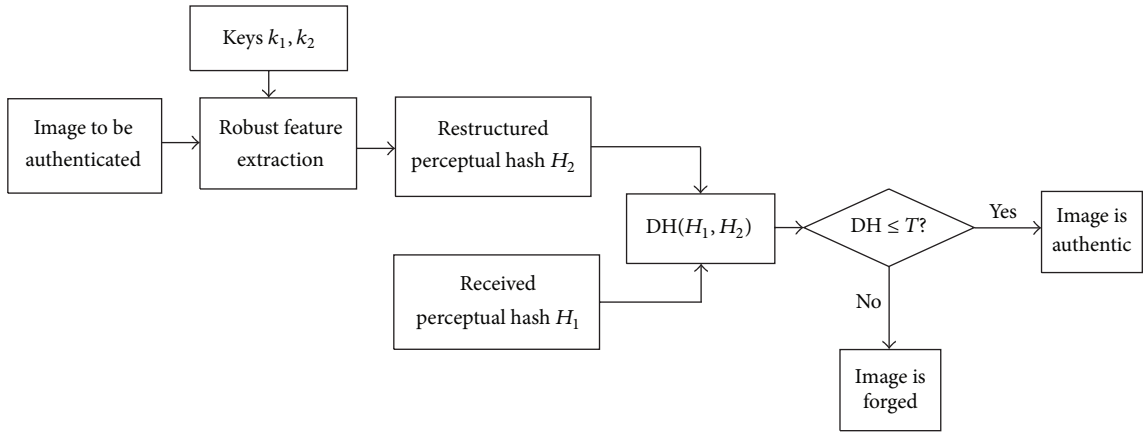


Figure 6: Image authentication process.

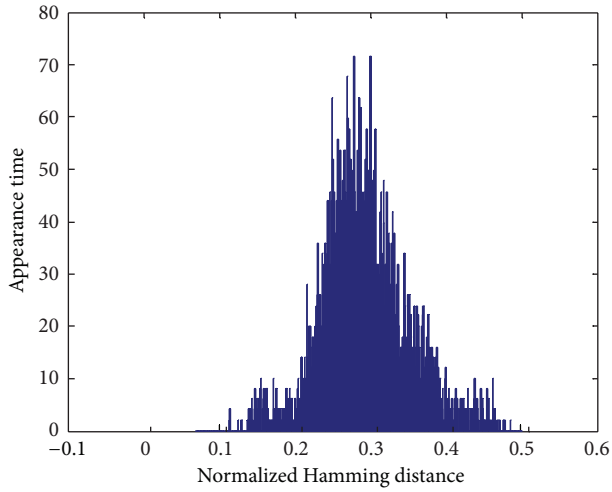


Figure 7: Visual fragility test.

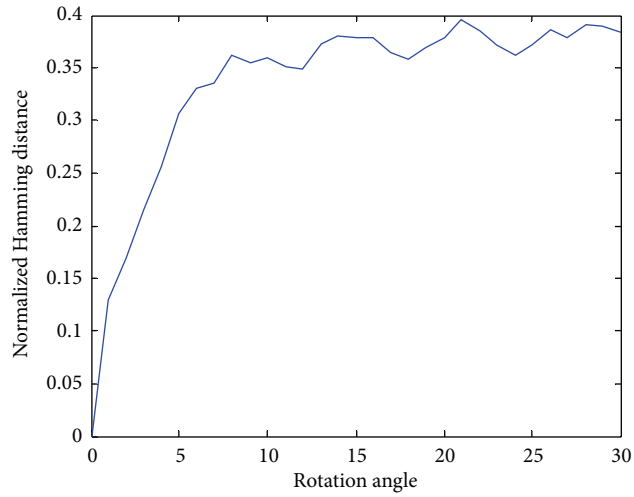


Figure 8: Robustness to image rotation.

where L is the length of perceptual Hashing sequence. If $DH(H_1, H_2) \leq T$, the image will be deemed an authentic version. Otherwise, if $DH(H_1, H_2) > T$, the image will be forged. The smaller the normalized Hamming distance is, the stronger the robustness is. Ideally, the normalized Hamming distance for the perceptually identical images is close to 0, while the normalized Hamming distance of two different images is close to 0.5.

5. Experimental Results and Analysis

5.1. Visual Fragility of Perceptual Hashes. The visual fragility represents that perceptually distinct images generate different perceptual hashes. We randomly select 80 images sized 300×300. The parameters are set as follows: $k_1 = 3.9$, $k_2 = 1.2$, $\Delta_1 = 10$, $\Delta_2 = 4$, and the number of random rectangle blocks is 150. Then we calculate their perceptual hashes and the Hamming distance between two perceptual hashing values $DH(H_1, H_2)$. Finally, 6320 matching results can be obtained. The statistical histogram distribution is shown in Figure 7. From Figure 7, we see the results can be approximate to the Gaussian distribution with the expectation $\mu = 0.2924$ and standard deviation $\sigma = 0.0574$. Setting the threshold $T = 0.125$, the conflict probability will be

$$P_c = 1 - \int_T^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2} dx = 2.8106 \times 10^{-6}. \quad (22)$$

As a result, the conflict probability is very small. Hence, the proposed perceptual hashing can ensure the visual fragility.

5.2. Perceptual Robustness. To test the perceptual robustness to geometric transforms, we rotate the "Lena" image sized 512×512 with different degrees and calculate the perceptual hashes. Compare the perceptual hashing value of rotated image with the original. The relationship of the rotation angle and the normalized Hamming distance is shown in Figure 8. When the image rotation angle is within 5°, the normalized Hamming distance is less than 0.3, so the proposed perceptual hashing algorithm is robust to image rotation within 5°.

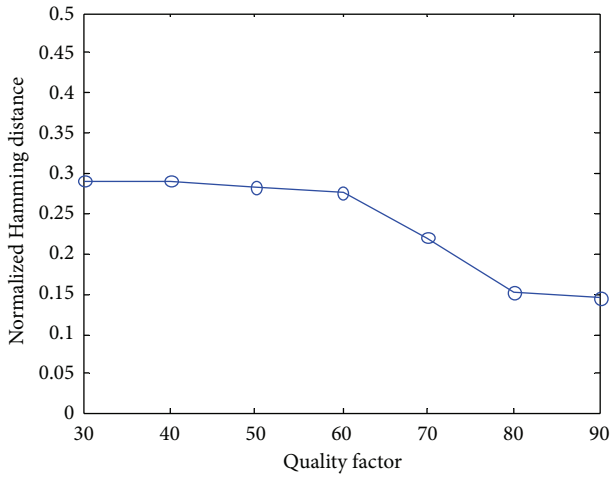


Figure 9: Robustness to image compression.

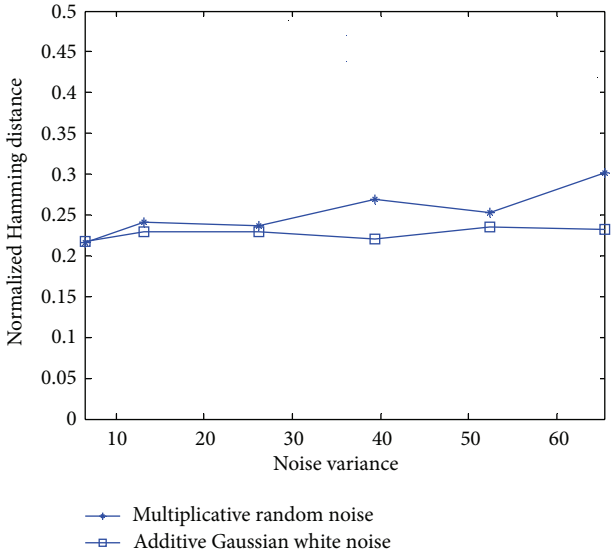


Figure 10: Robustness to noise blurring.

Figure 9 shows the robustness to image compression. When the quality factor of JPEG compression is changed from 90 to 30, the normalized Hamming distance between the original image and the compressed image is less than 0.3, so the proposed perceptual hashing algorithm is robust to image compression. The smaller the quality factor is, the larger the compression degree is. From Figure 9, we see that the robustness is getting more and more strong with the accretion of the quality factor.

Figure 10 shows the robustness to additive Gaussian white noise and uniformly distributed multiplicative random noise. When the image is blurred by different noise degrees with variance 6.554 ~ 65.54, we calculate the perceptual hashes. From Figure 10, we see that, when the variance is less than 65.54, the normalized Hamming distance between the original image and the noise blurred image is less than 0.3; That is to say, the proposed algorithm is robust to Gaussian white noise and uniformly distributed random multiplicative noise during image transmission and storage.

Table 1: Perceptual hashing segments by different keys.

Segment 1 ($k_1 = 3.9, k_2 = 1.2$)	$\cdots$1001011100011011000011010000$\cdots$
Segment 2 ($k_1 = 3.91, k_2 = 1.2$)	$\cdots$1000110110010100100110000100$\cdots$
Segment 3 ($k_1 = 3.9, k_2 = 1.21$)	$\cdots$1010100110001001110010011110$\cdots$

5.3. Security. Because the chaotic sequence is nonperiodic and sensitive to the initial value, the chaotic maps (6) and (7) are used to generate the pseudorandom sequence in this paper. Then, the digital image is randomly divided into overlapping rectangular regions by the chaotic sequence for perceptual image hashing extraction. Thus, if chaotic initial values are changed, that is, the keys are different, the extracted perceptual image hashing will be different. Table 1 lists the segments of the generated perceptual image hashes by different keys, which indicates the perceptual hashing is intractable without the secret key. We can calculate that the normalized Hamming distance is 0.4167 between the corresponding locations of the perceptual hashing segment 1 and segment 2 and that it is 0.4283 between segment 1 and segment 3. The two normalized Hamming distances are all close to 0.5, so the security meets the application requirement. Therefore, without knowing the key, even if the perceptual image hashing algorithm is known, the correct perceptual hashing value generated by the image cannot be leaked.

6. Conclusions

In this paper, we have proposed a robust image authentication methodology for authenticity, creditability, and integrity transmission and storage of authenticated images based on the perceptual hashing technique in WMSNs. First, a gravity center-based perceptual image hashing algorithm is proposed with compactness, perceptual robustness, visual fragility, and security. The generated perceptual hashing value is a short binary string as a digest of the image in order to tackle the problem of severe energy constraints and perceptual image redundancy in WMSNs. Furthermore, a distributed processing strategy for perceptual image hashes is developed to meet the limited computing resources of sensor nodes in WMSNs. The experimental results demonstrate that our scheme has the satisfactory authentication performance for perceptually distinct and identical images.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (NSFC) under Grant no. 61170226, the Fundamental Research Funds for the Central Universities under Grants nos. SWJTU11CX047, SWJTU12ZT02, the Young Innovative Research Team of Sichuan Province under Grant no. 2011JTD0007, and Chengdu Science and Technology program under Grant no. 12DXYB214JH-002.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 32–39, 2007.

[2] H. Wang, M. Hempel, D. Peng, W. Wang, H. Sharif, and H. H. Chen, "Index-based selective audio encryption for wireless multimedia sensor networks," *IEEE Transactions on Multimedia*, vol. 12, no. 3, pp. 215–223, 2010.

[3] Y. Y. Zhang, X. Z. Li, J. M. Liu, J. C. Yang, and B. J. Cui, "A secure hierarchical key management scheme in wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 547471, 8 pages, 2012.

[4] S. Han, E. Chang, L. Gao, and D. Tharam, "Taxonomy of attacks on wireless sensor networks," in *Proceedings of the 1st European Conference on Computer Network Defense*, pp. 97–105, Glamorgan, UK, December 2005.

[5] S. G. Yoo, K. Y. Park, and J. Kim, "A security-performance-balanced user authentication scheme for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 382810, 11 pages, 2012.

[6] B. Harjito and S. Han, "Wireless multimedia sensor networks applications and security challenges," in *Proceedings of the 5th International Conference on Broadband Wireless Computing, Communication and Applications (BWCCA '10)*, pp. 842–846, November 2010.

[7] M. K. Khan, "Fingerprint biometric-based self-authentication and deniable authentication schemes for the electronic world," *IETE Technical Review*, vol. 26, no. 3, pp. 191–195, 2009.

[8] Z. Li, Q. Sun, Y. Lian, and C. W. Chen, "Joint source-channel-authentication resource allocation and unequal authenticity protection for multimedia over wireless networks," *IEEE Transactions on Multimedia*, vol. 9, no. 4, pp. 837–850, 2007.

[9] W. Wang, D. Peng, H. Wang, H. Sharif, and H. H. Chen, "Energy-distortion-authentication optimized resource allocation for secure wireless image streaming," in *Proceedings of the IEEE Conference on Wireless Communications and Networking (WCNC '08)*, pp. 2810–2815, April 2008.

[10] Z. Zhang, Q. Sun, W. C. Wong, J. Apostolopoulos, and S. Wee, "An optimized content-aware authentication scheme for streaming JPEG-2000 images over lossy networks," *IEEE Transactions on Multimedia*, vol. 9, no. 2, pp. 320–331, 2007.

[11] W. Wang, D. Peng, H. Wang, H. Sharif, and H. H. Chen, "A multimedia quality-driven network resource management architecture for wireless sensor networks with stream authentication," *IEEE Transactions on Multimedia*, vol. 12, no. 5, pp. 439–447, 2010.

[12] Z. Zhang, Q. Sun, W. C. Wong, J. Apostolopoulos, and S. Wee, "Rate-distortion-authentication optimized streaming of authenticated video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 5, pp. 544–557, 2007.

[13] Q. Sun and S. F. Chang, "A secure and robust digital signature scheme for JPEG2000 image authentication," *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 480–494, 2005.

[14] Q. Sun, D. He, and Q. Tian, "A secure and robust authentication scheme for video transcoding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 10, pp. 1232–1244, 2006.

[15] D. Skraparlis, "Design of an efficient authentication method for modern image and video," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 2, pp. 417–426, 2003.

[16] X. D. Lv and Z. J. Wang, "Perceptual image hashing based on shape contexts and local feature points," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1081–1093, 2012.

[17] M. Tagliasacchi, G. Valenzise, and S. Tubaro, "Hash-based identification of sparse image tampering," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2491–2504, 2009.

[18] K. R. Castleman, *Digital Image Processing*, Prentice Hall, New York, NY, USA, 1998.

[19] G. Wang, D. Kim, and G. Cho, "A secure cluster formation scheme in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 301750, 14 pages, 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 730831, 7 pages
http://dx.doi.org/10.1155/2013/730831

Research Article

A New User Authentication Protocol for Wireless Sensor Networks Using Elliptic Curves Cryptography

Wenbo Shi[1] and Peng Gong[2]

[1] *Department of Electronic Engineering, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China*
[2] *National Key Laboratory of Mechatronic Engineering and Control, School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China*

Correspondence should be addressed to Wenbo Shi; swb319@hotmail.com

Received 12 January 2013; Revised 16 March 2013; Accepted 16 March 2013

Academic Editor: Muhammad Khurram Khan

User authentication in wireless sensor networks (WSNs) is a critical security issue due to their unattended and hostile deployment in the field. Since sensor nodes are equipped with limited computing power, storage, and communication modules, authenticating remote users in such resource-constrained environments is a paramount security concern. To overcome the weaknesses of Yeh et al.'s protocol, we proposed a new authentication protocol for wireless sensor networks using elliptic curves cryptography. The comparisons show that our protocol is more suitable for WSNs.

1. Introduction

Wireless sensor networks (WSNs) are becoming more and more popular in everyday life as they offer economically viable, real-time monitoring solutions. These wireless sensors can be quickly and easily deployed in hostile environments, and WSNs are now widely used in a variety of real-time applications, such as vehicular tracking, habitat monitoring, environment control, military surveillance, healthcare monitoring, wildlife monitoring, and traffic monitoring. One recent survey declared that, in the near future, WSNs will become an intelligent and integral part of daily lives [1].

A WSN consists of a discrete group of independent, low cost, and low power nodes with limited memory and computation power. They communicate wirelessly over limited frequency and low bandwidth [1]. More specifically, sensor nodes collectively monitor the area and sense substantial amounts of data, which are transmitted to the base station traversing some nodes via RF signals and routing schemes.

A key requirement for WSN is user authentication [2, 3]. The client devices (remote wireless sensor nodes) need to be authenticated before being allowed to join the WSN and have access to the WSN's resources. To date, most user authentication methods have focused on protocol implementations in the network and link layers. It should be noted that, in order to limit power consumption by sensor nodes and to overcome limitations in computation capacity, user authentication in a WSN is typically done in dedicated gateway node (GW node) [1].

In 2004, Sastry and Wagner [4] proposed a security enhancement using access control lists (ACLs) in the GW node. In Sastry and Wagner's protocol, an ACL would be maintained besides the client's identity and the arranging of the nearest sensor node. Watro et al. [5] proposed a user authentication protocol employing RSA and Diffie-Hellman algorithms, but this protocol is open to hostile attack by a user masquerading as a sensor node. Wong et al. [6] proposed a dynamic user authentication protocol using hash function. Das [7] and Tseng et al. [8] demonstrated that both Watro's and Wong's user authentication methods were vulnerable to stolen-verifier, replay, and forgery attacks. To improve the security, Das [7] proposed a two-factor user authentication protocol. In 2007, Tseng et al. [8] show that Wong's protocol was vulnerable to stolen passwords. Tseng et al. also proposed an enhanced user authentication protocol to improve overcome the weakness. However, Khan and Alghathbar [9, 10] show that Das' protocol did not provide mutual authentication between gateway node and

sensor node and was vulnerable to gateway node bypassing attack and privileged-insider attack. Chen and Shih [11] also demonstrated that Das' protocol did not provide mutual authentication between gateway node and sensor node. Chen and Shih [11] also proposed a more secure and robust two-factor user authentication in WSNs. Unfortunately, Yeh et al. [12] found that Chen and Shih's protocol failed to provide a secure method for updating user passwords and was vulnerable to the insider attack problem. To improve the performance and the security, Yeh et al. [12] proposed the first user authentication protocol for WSNs using the elliptic curve cryptography (ECC). ECC was first proposed by Miller [13] and Koblitz [14], and its security was based upon the difficulty of elliptic curve discrete logarithm problem. Compared with the other cryptography, ECC offers a better performance because it can achieve the same security with a smaller key size. For example, 160-bit ECC and 1024-bit RSA have the same security level in practice [15]. Thus, ECC-based authentication schemes are very suitable for WSNs.

Unfortunately, Han [16] found that the Yeh et al. protocol had the following weaknesses: (1) no mutual authentication between the user and the sensor node, (2) no perfect forward secrecy, and (3) no key agreement between the user and the sensor node. To overcome the weaknesses of Yeh et al.'s protocol, we propose a new ECC-based user authentication protocol for WSNs.

The remainder of this paper is organized as follows. In Section 2, we propose our ECC-based authentication protocol for WSNs. The security analysis of the proposed protocol is presented in Section 3. In Section 4, performance analysis is presented. Conclusions are given in Section 5.

2. The Proposed Protocol

To solve the weakness of Yeh et al.'s scheme, we propose a new ECC-based user authentication protocol for WSNs. Thus, before issuing a query to a sensor node, each user must register with the gateway in a secure manner so that they can access the real-time sensors' data. Upon the successful user registration request, the gateway node personalizes a smart card for every registered user. Then, a user can submit his query in an authentic way and access the sensor network data at any time within an administratively configurable period [6].

In order to execute the proposed framework, we considered that the gateway is a trusted node and it holds two master keys (x and y), which are sufficiently large for the sensor network. Before starting the system, it is assumed that the gateway and the sensor nodes share a long-term common secret key, that is, $\mathrm{SK}_{GS} = h(S_n||y)$ using any key agreement protocol. For example, Watro et al. [17] demonstrated that, with the careful design, D-H key agreement protocol [18] can be easily deployed on most constrained devices. Here, $h(\cdot)$ is a collision-free one-way hash function (i.e., SHA-1), which has an output length of 160 bits [19] and is used throughout this paper.

It is assumed that some identical secure symmetric cryptosystems are publicly available and stored in the gateway and the sensor node. As a result only the users registered with the gateway have access privileges to the sensors, which share a long-term secret with the gateway. The framework is divided into four phases, namely, user registration phase, login phase, authentication phase, and password update phase. For convenience, the notations used throughout this paper are summarized as follows:

p, n: two large prime numbers;

F_p: a finite field;

E: an elliptic curve defined on finite field F_p with large order;

G: the group of elliptic curve points on E;

P: a point on elliptic curve E with order n;

U: a user;

ID_U: the user U identity;

pw_U: the user U password;

GW node: the gateway node of WSN;

S_n: a sensor node of WSN;

ID_{S_n}: the sensor node S_n identity;

x, y: the master keys of GW node;

$h(\cdot)$: a secure one-way hash function;

$||$: a string concatenation operation;

$\oplus$: a string XOR operation;

ECDLP: the discrete logarithm problem, that is, given $Q \in G$ to compute $x \in Z_n^*$ such that $Q = xP$;

ECCDHP: the computational Diffie-Hellman problem, that is, given $aP, bP \in G$ to compute abP.

2.1. Registration Phase. In this phase, user U has to submit an identity, ID_U, and a password, pw_U, to the GW node in a secured way. Then, the GW node issues a license to U. The detailed steps are depicted as follows.

(1) U chooses his identity ID_U and password pw_U, generates a random number b_U, and computes $\overline{\mathrm{pw}}_U = h(\mathrm{pw}_U \oplus b_U)$. Then, U sends ID_U and $\overline{\mathrm{pw}}_U$ to the GW node.

(2) Upon receiving the registration request, GW node computes $K_U = h(\mathrm{ID}_U||x) \times P$, $B_U = h(\mathrm{ID}_U \oplus \overline{\mathrm{pw}}_U)$, and $W_U = h(\mathrm{ID}_U||\overline{\mathrm{pw}}_U) \oplus K_U$. Then GW node stores $\{B_U, W_U, h(\cdot)\}$ into a smart card and sends it to the user U.

(3) After receiving the smart card, the user U inputs b_U into it and finishes the registration.

2.2. Login Phase. When U enters an ID_U and a pw_U in order to deliver some query to or access data from the WSN, the smart card must perform the following steps to validate the legitimacy of U. Figure 1 shows both the login phase and the authentication phase.

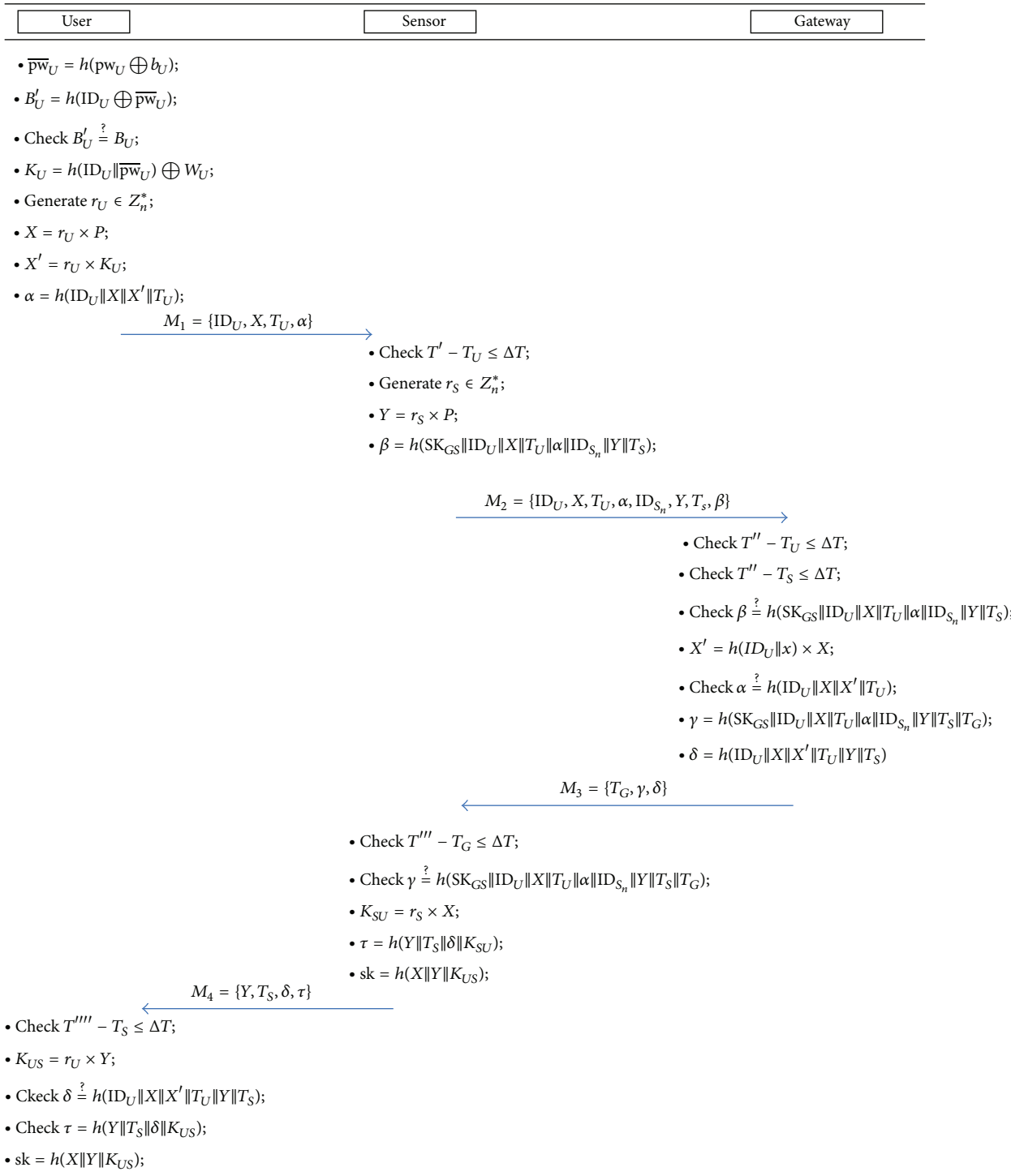


Figure 1: The login phase and the authentication phase of our scheme.

(1) User U inserts his smart card into the terminal and enters his identity ID_U and password pw_U.

(2) The smart card computes $\overline{\text{pw}}_U = h(\text{pw}_U \oplus b_U)$ and $B'_U = h(\text{ID}_U \oplus \overline{\text{pw}}_U)$ and checks whether $B'_U = B_U$. If it does not hold, the smartcard stops the request. Otherwise, the smart card computes $K_U = h(\text{ID}_U || \overline{\text{pw}}_U) \oplus W_U$. Then the smart card generates a random number $r_U \in Z_n^*$ and computes $X = r_U \times P$, $X' = r_U \times K_U$, and $\alpha = h(\text{ID}_U || X || X' || T_U)$, where T_U is the current timestamp of U's system. At last, the smart card sends the login message $M_1 = \{\text{ID}_U, X, T_U, \alpha\}$ to S_n.

2.3. Authentication Phase. After receiving the login request message M_1 at time T', the sensor node S_n executes the

following steps to authenticate U's requests by the following steps.

(1) S_n checks whether $T' - T_U \leq \Delta T$ holds, where ΔT is the legal time interval for transmission delay. If the answer is yes, the validity of T_U can be assured, and S_n proceeds to the next step. If no, the S_n rejects the request.

(2) S_n generates a random number $r_S \in Z_n^*$ and computes $Y = r_S \times P$ and $\beta = h(\mathrm{SK}_{GS}||\mathrm{ID}_U||X||T_U||\alpha||\mathrm{ID}_{S_n}||Y||T_S)$, where T_S is the current timestamp of S_n's system. At last, S_n sends $M_2 = \{\mathrm{ID}_U, X, T_U, \alpha, \mathrm{ID}_{S_n}, Y, T_S, \beta\}$ to GW node.

After receiving the message M_2 at time T'', GW node performs the following actions.

(1) GW node checks whether $T'' - T_U \leq \Delta T$ and $T'' - T_S \leq \Delta T$ hold, where ΔT is the legal time interval for transmission delay. If the answer is yes, the validity of T_U and T_S can be assured, and GW node proceeds to the next step. If no, GW node rejects the request.

(2) GW node uses long-term key SK_{GS} to check whether the equation $\beta = h(\mathrm{SK}_{GS}||\mathrm{ID}_U||X||T_U||\alpha||\mathrm{ID}_{S_n}||Y||T_S)$ holds. If the equation does not hold, GW node stops the session. Otherwise, GW node computes $X' = h(\mathrm{ID}_U||x) \times X$ using his master key x. Then, GW node checks whether the equation $\alpha = h(\mathrm{ID}_U||X||X'||T_U)$ holds. If the equation does not hold, GW node stops the session. Otherwise, GW node computes $\gamma = h(\mathrm{SK}_{GS}||\mathrm{ID}_U||X||T_U||\alpha||\mathrm{ID}_{S_n}||Y||T_S||T_G)$ and $\delta = h(\mathrm{ID}_U||X||X'||T_U||Y||T_S)$. At last, GW node sends the message $M_3 = \{T_G, \gamma, \delta\}$ to S_n.

After receiving the message M_3 at time T''', S_n performs the following actions to authenticate U and GW node.

(1) S_n checks whether $T''' - T_G \leq \Delta T$ holds, where ΔT is the legal time interval for transmission delay. If the answer is yes, the validity of T_G can be assured, and S_n proceeds to the next step. If no, S_n rejects the request.

(2) S_n uses long-term key SK_{GS} to check whether the equation $\gamma = h(\mathrm{SK}_{GS}||\mathrm{ID}_U||X||T_U||\alpha||\mathrm{ID}_{S_n}||Y||T_S||T_G)$ holds. If the equation does not hold, S_n stops the session. Otherwise, S_n computes $K_{SU} = r_S \times X$, $\tau = h(Y||T_S||\delta||K_{SU})$, and the session key $\mathrm{sk} = h(X||Y||K_{SU})$ and sends $M_4 = \{Y, T_S, \delta, \tau\}$ to the smart card.

After receiving the message M_4 at time T'''', the smart card performs the following actions to authenticate S_n.

(1) The smart card checks whether $T'''' - T_S \leq \Delta T$ holds, where ΔT is the legal time interval for transmission delay. If the answer is yes, the validity of T_S can be assured, and the smart card proceeds to the next step. If no, the smart card rejects the request.

(2) The smart card computes $K_{US} = r_U \times Y$ and checks whether the equations $\delta = h(\mathrm{ID}_U||X||X'||T_U||Y||T_S)$ and $\tau = h(Y||T_S||\delta||K_{US})$ hold. If either of the two equations does not hold, the smart card stops the session. Otherwise, S_n is authenticated and the smart card computes the session key $\mathrm{sk} = h(X||Y||K_{US})$.

2.4. Password Update Phase. The password update phase is invoked whenever user U wants to update his old password pw_U. The password update phase is described below.

(1) User U inserts his smart card into the terminal and enters his identity ID_U, the old password pw_U, and the new password pw'_U.

(2) The smart card computes $\overline{\mathrm{pw}}_U = h(\mathrm{pw}_U \oplus b_U)$ and $B'_U = h(\mathrm{ID}_U \oplus \overline{\mathrm{pw}}_U)$ and checks whether $B'_U = B_U$. If it does not hold, the smart card stops the request. Otherwise, the smart card computes $K_U = h(\mathrm{ID}_U||\overline{\mathrm{pw}}_U) \oplus W_U$, $\overline{\mathrm{pw}}'_U = h(\mathrm{pw}'_U \oplus b_U)$, and $W'_U = K_U \oplus h(\mathrm{ID}_U||\overline{\mathrm{pw}}'_U)$. At last, the smart card replaces W_U with W'_U.

3. Security Analysis

In this section, we will discuss the security of our protocol as follows.

Mutual Authentication. Our scheme provides mutual authentication, where all entities (i.e., user, gateway, and sensor nodes) are mutually authenticating each other. More specifically, when the GW node receives the message $M_2 = \{\mathrm{ID}_U, X, T_U, \alpha, \mathrm{ID}_{S_n}, Y, T_S, \beta\}$, it can make sure that the user message $M_1 = \{\mathrm{ID}_U, X, T_U, \alpha\}$ is included in the sensor node message M_2. When the sensor node receives message $M_3 = \{T_G, \gamma, \delta\}$, it ensures that this message is generated by the GW node. Furthermore, when the user receives message $M_4 = \{Y, T_S, \delta\}$, he can also confirm that this message is generated by the sensor node. Hence, mutual authentication is achieved.

Replay Attacks. Our scheme is resistant to replay attacks, because the authenticity of messages M_1, M_2, M_3, and M_4 is validated by checking the freshness of four timestamps. Let us assume an intruder intercepts a login request message $M_1 = \{\mathrm{ID}_U, X, T_U, \alpha\}$ and attempts to access the sensor node by replaying the same message M_1. The verification of this login attempt fails since the time difference expires (i.e., $T' - T_U > \Delta T$). Similarly, if an intruder intercepts a valid message $M_2 = \{\mathrm{ID}_U, X, T_U, \alpha, \mathrm{ID}_{S_n}, Y, T_S, \beta\}$ and attempts to replay it to the GW node, the verification request will fail at the GW node because the time difference expires again (i.e., $T'' - T_S > \Delta T$). Thus, our protocol is secure against replaying of messages.

User Impersonation Attacks. An attacker cannot impersonate the user. Suppose an attacker forges a login message $M_1 = \{\mathrm{ID}_U, X, T_U, \alpha\}$. Now, he will again try to login into the system with the modified message $\{\mathrm{ID}_U, X^*, T'_U, \alpha^*\}$. However, the

Table 1

	Yeh et al. 's protocol	Our protocol
Computational cost (user)	$2TG_{\text{mul}} + 1TG_{\text{grp}} + 1TG_{\text{add}} + 4TG_h \approx 2.5TG_{\text{mul}}$	$3TG_{\text{mul}} + 5TG_h \approx 3TG_{\text{mul}}$
Computational cost (sensor node)	$2TG_{\text{mul}} + 1TG_{\text{grp}} + 1TG_{\text{add}} + TG_{\text{mtph}} + 1TG_h \approx 3TG_{\text{mul}}$	$2TG_{\text{mul}} + 3TG_h \approx 2TG_{\text{mul}}$
Computational cost (gateway)	$3TG_{\text{mul}} + 1TG_{\text{grp}} + TG_{\text{mtph}} + 1TG_h \approx 3TG_{\text{mul}}$	$1TG_{\text{mul}} + 4TG_h \approx 1TG_{\text{mul}}$
Mutual authentication	No	Yes
Perfect forward secrecy	No	Yes
Key agreement between user and node	No	Yes
User impersonation attack	No	No
Sensor node impersonation attack	Yes	No
Gateway impersonation attack	No	No
Inside attack	No	No
Replay attacks	No	No
Stolen-verifier attacks	No	No
Man-in-the-middle attack	No	No

attacker cannot forge α^* without knowing $K_U = h(\text{ID}_U||x)\times P$ or the master key x since he will be faced with ECDLP. Therefore, it is not possible to impersonate the user.

Sensor Impersonation Attacks. As long as an attacker does not know the secret key SK_{GS}, he cannot generate a legal message $M_2 = \{\text{ID}_U, X, T_U, \alpha, \text{ID}_{S_n}, Y, T_S, \beta\}$. Then he cannot cheat the gateway. At the same time, he cannot generate a legal message $M_4 = \{Y, T_S, \delta\}$ without knowing the master key x. Therefore, it is not possible to impersonate the sensor.

Gateway Impersonation Attacks. As long as an attacker does not possess the secret key SK_{GS}, he cannot impersonate the gateway and cannot cheat the sensor node. Hence, it frustrates attackers to generate the valid message M_4 to the sensor node. Therefore, it is not possible to impersonate the gateway.

Man-in-the-Middle Attack. Man-in-the-middle attack means that an active attacker intercepts the communication line between a legal user and the server and uses some means to successfully masquerade as both the server to the user and the user to the server. Then, the user will believe that he is talking to the intended server and vice versa. From the above discussion we know that our protocol can provide mutual authentication, and then the "man-in-the-middle" attack can be resisted.

Stolen-Verifier Attacks. An attacker who steals the password verifier (e.g., hashed passwords) from the gateway can use the stolen verifier to impersonate a legal user to login to the system. The proposed scheme is free from the stolen verifier attack. There is no such information stored at the server, by which an adversary can make a fabricated login request to impersonate a legal user to login the server or can impersonate the gateway to cheat the legal user and the sensor node.

Insider Attacks. It is possible in a real-time environment, when the gateway manager or system administrator can use the user password pw_U (e.g., weak password), to impersonate the user U through any other network gateways. In this case, our scheme does not give any room for privileged insiders, since, in the registration phase, the user U is passing $\overline{\text{pw}}_U = h(\text{pw}_U \oplus b_U)$ instead of the plain password. Thus, the insider of the GW node cannot get pw_U easily. Here, b_U is a sufficiently high entropy number, which is not revealed to the GW node. Furthermore, the proposed scheme does not store any verifier table and can resist the insider attacks.

Perfect Forward Secrecy. A protocol is said to be *perfect forward secrecy* if compromise of the three private keys of the participating entities does not affect the security of the previous session keys. Two aspects are related to this notion, that is, perfect forward secrecy (p-FS) and master key perfect forward secrecy. p-FS means that the compromise of both user's and sensor node's long-term private keys would not affect the secrecy of the previously established session keys. Master key p-FS is satisfied if the session key secrecy still holds even when the server's master key is compromised. Our protocol satisfies both p-FS and master key p-FS by using $\text{sk} = h(X||Y||r_S \times X)$ or $\text{sk} = h(X||Y||r_U \times Y)$ as the shared secret. If user's private keys or gateway's master key is compromised, the adversary cannot compute r_U or r_S from X and Y since he has to solve the ECCDHP, thus satisfying both p-FS and master key p-FS.

4. Performance Comparison

For the convenience of evaluating the computational cost, we define some notations as follows.

TG_{mul}: the time of executing a scalar multiplication operation of point.

TG_{add}: the time of executing an addition operation of points.

TG_{grp}: the time of generating a random number point.

TG_{mtph}: the time of executing a map-to-point hash function.

TG_h: the time of executing a one-way hash function.

In Table 1, we summarize the performance results of the proposed protocol. In Table 1, we know that the user, the sensor node, and the gateway require $3TG_{\text{mul}} + 5TG_h$, $2TG_{\text{mul}} + 3TG_h$, and $1TG_{\text{mul}} + 4TG_h$, separately. From the theoretical analysis [20] and the experimental result [21, 22], we know that the relative computation cost of generating a random number point and executing a map-to-point hash function is about 1/2 times that of the scalar multiplication.

Besides, the computation costs of TG_{mul} are considerably higher than TG_{add} and TG_h. Then, the computational costs of the user, the sensor node, and the gateway in Yeh et al.'s protocol are about $3TG_{\text{mul}}$, $3TG_{\text{mul}}$, and $3TG_{\text{mul}}$, separately. The computational costs of the user, the sensor node, and the gateway in our protocol are about $3TG_{\text{mul}}$, $2TG_{\text{mul}}$, and $1TG_{\text{mul}}$, separately. Then our protocol has better performance at the sensor node side and the gateway side. Moreover, Yeh et al.'s protocol cannot provide (1) mutual authentication between the user and the sensor node, (2) perfect forward secrecy, and (3) key agreement between the user and the sensor node, and then our protocol enhances the security at the cost of increasing user's computation cost slightly.

5. Conclusions

This paper provides a new ECC-based user authentication protocol for WSNs. The proposed protocol performs more efficiently in terms of computation cost, communication cost, and security. Compared with the protocol of Yeh et al., the proposed protocol in this paper can prevent general security issues and provide mutual authentication to protect inside security and outside security. Therefore, the proposed protocol is more suited to WSNs environments.

Acknowledgments

The authors thank the editors and the anonymous reviewers for their valuable comments. This research was supported by National Natural Science Foundation of China (nos. 61202447 and 61201180), Natural Science Foundation of Hebei Province of China (no. F2013501066), Northeastern University at Qinhuangdao Science and Technology Support Program (no. xnk201307), Beijing Natural Science Foundation (no.4132055), and Excellent Young Scholars Research Fund of Beijing Institute of Technology.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramamiam, and E. A. Cayirci, "Survey on saensor network," *IEEE Communications Magazine*, vol. 40, pp. 102–114, 2002.

[2] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic ID-based remote user authentication scheme," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 629–631, 2004.

[3] K. C. Leung, L. M. Cheng, A. S. Fong, and C. K. Chan, "Cryptanalysis of a modified remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1243–1245, 2003.

[4] N. Sastry and D. Wagner, "Security considerations for IEEE 802.15.4 networks," in *Proceedings of the ACM Workshop on Wireless Security (WiSe '04)*, pp. 32–42, Philadelphia, Pa, USA, October 2004.

[5] R. Watro, D. Kong, S. F. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, pp. 59–64, Washington, DC, USA, October 2004.

[6] K. H. M. Wong, Z. Yuan, C. Jiannong, and W. Shengwei, "A dynamic user authentication scheme for wireless sensor networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pp. 244–251, Taichung, Taiwan, June 2006.

[7] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.

[8] H. R. Tseng, R. H. Jan, and W. Yang, "An improved dynamic user authentication scheme for wireless sensor networks," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 986–990, Washington, DC, USA, November 2007.

[9] M. K. Khan and K. Alghathbar, "Security analysis of two-factor authentication in wireless sensor networks," in *Proceedings of the Advances in Computer Science and Information Technology (AST/UCMA/ISA/ACN '10)*, pp. 55–60, Miyazaki, Japan, June 2010.

[10] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'," *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.

[11] T. H. Chen and W. K. Shih, "A robust mutual authentication protocol for wireless sensor networks," *ETRI Journal*, vol. 32, no. 5, pp. 704–712, 2010.

[12] H. L. Yeh, T. H. Chen, P. C. Liu, T. H. Kim, and H. W. Wei, "A secured authentication protocol for wireless sensor networks using Elliptic Curves Cryptography," *Sensors*, vol. 11, no. 5, pp. 4767–4779, 2011.

[13] V. S. Miller, "Use of elliptic curves in cryptography," in *Proceedings of the Advances in Cryptology (CRYPTO '85)*, vol. 218 of *Lecture Notes in Computer Science*, pp. 417–426, Springer, 1986.

[14] N. Koblitz, "Elliptic curve cryptosystem," *Mathematics of Computation*, vol. 48, pp. 203–209, 1987.

[15] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Lecture Notes in Computer Science, Springer, New York, NY, USA, 2004.

[16] W. Han, "Weakness of a Secured Authentication Protocol for Wireless Sensor Networks Using Elliptic Curves Cryptography," http://eprint.iacr.org/2011/293 .

[17] R. Watro, D. Kong, S. F. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04)*, pp. 59–64, October 2004.

[18] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[19] "National Institute of Standards and Technology, FIPS PUB 180-1, Secure Hash Standard," http://www.techheap.com/cryptography/hash/fip180-1.pdf.

[20] L. Chen, Z. Cheng, and N. P. Smart, "Identity-based key agreement protocols from pairings," *International Journal of Information Security*, vol. 6, no. 4, pp. 213–241, 2007.

[21] X. Cao, X. Zeng, W. Kou, and L. Hu, "Identity-based anonymous remote authentication for value-added services in mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3508–3517, 2009.

[22] H. Debiao, C. Jianhua, and H. Jin, "An ID-based client authentication with key agreement protocol for mobile client-server environment on ECC with provable security," *Information Fusion*, vol. 13, no. 3, pp. 223–230, 2012.