Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 718286, 11 pages
http://dx.doi.org/10.1155/2014/718286

Research Article

Online Authentication and Key Establishment Scheme for Heterogeneous Sensor Networks

Sarmad Ullah Khan,[1] Luciano Lavagno,[2] Claudio Pastrone,[3] and Maurizio A. Spirito[3]

[1] *Electrical Department, CECOS University, Peshawar 25000, Pakistan*
[2] *Electronics and Telecommunication Department, Politecnico di Torino, 10129 Torino, Italy*
[3] *Istituto Superiore Mario Boella (ISMB), Pervasive Radio Technologies (PeRT) Lab, 10138 Torino, Italy*

Correspondence should be addressed to Sarmad Ullah Khan; sarmadullahkhan1@gmail.com

Received 11 June 2014; Accepted 16 October 2014; Published 18 November 2014

Academic Editor: Alessandro Nordio

In recent years, the adaptation of wireless sensor networks (WSNs) to application areas requiring mobility increased the security threats against confidentiality, integrity, and privacy of the information as well as against their connectivity. Since key management plays an important role in securing both information and connectivity, a proper authentication and key management scheme is required in mobility enabled applications where the authentication of a node with the network is a critical issue. In this paper, we present an authentication and key management scheme supporting node mobility in a heterogeneous WSN that consists of several mobile sensor nodes and a few fixed sensor nodes. We analyze our proposed solution by using the OMNET++ simulator to show that it requires less memory space and provides better connectivity and network resilience against node capture attacks compared to some existing schemes. We also propose two levels of secure authentication methods for the mobile sensor nodes for secure authentication and key establishment.

1. Introduction

The wireless sensor network (WSN) was initially considered to be used for military applications but the popularity of WSN, because of its small size, low cost, and being easy to deploy and manage, makes it used in a variety of applications. This opens a door to new research challenges to the research community. Since sensor nodes are usually deployed in possibly remote and unattended locations, they are definitely prone to security attacks. Hence to secure the network operation and securely gather and forward the information, security threats and their countermeasures should be considered at design time in terms of both requirements and implementation techniques. However, such tasks are not trivial due to the limited energy, computation, memory, and bandwidth resources available in sensor nodes. Fundamentally, any practical WSN security countermeasure must be, on the one hand, secure enough to satisfy the initial requirements and, on the other hand, lightweight enough not to interfere with normal WSN operations.

The design of security algorithms considering the homogeneous sensor networks where all nodes have the same capabilities (memory, radio range, computational power, battery life, etc.) was the first step to secure sensor networks. However, some research work has shown, both theoretically [1–3] and through simulation experiments and test bed measurements [4], that homogeneous sensor networks have high communication and computation overheads and high storage requirements and suffer from severe performance bottlenecks. Hence, recent research work [5–9] introduced heterogeneous sensor networks, which consists of a small number of powerful high-end sensors (H-sensors) and large number of low-end sensors (L-sensors). To achieve better performance and scalability, H-sensors have more resources in terms of energy, computation power, storage capacity, and transmission power, and so forth compared to L-sensors. However, both H-Sensors and L-sensors are still highly vulnerable in nature and are exposed to several security threats and particularly prone to physical attacks. Thus, proper security mechanisms should be applied to protect these nodes against specific attacks including Denial of Service (DoS) attacks, Sybil attacks, compromised node attacks, node replication attacks, and physical tampering. Physical attacks become even more troublesome when the nodes are mobile and the possible intruder can more easily target them. Most security features to provide secure communication and authentication essentially rely on encryption. Nevertheless, encryption is not possible in such a distributed environment without adequate key management mechanisms. For this reason, the adoption of a suitable key management scheme plays a central role in the definition of a secure WSN.

1.1. Key Ideas of the Proposed Approach. A key management scheme for heterogeneous wireless sensor network is proposed to overcome the scalability issues by providing almost 100% network connectivity, reducing memory cost while increasing the network resilience against attacks, and reducing communication overhead to save the energy and increase of network lifetime.

Hence, a novel key management scheme for heterogeneous sensor networks suitable for scenarios with partial mobility is presented. The proposed solution relies on two types of keys: authentication keys and secret communication codes used to generate secret keys whenever needed. The key material is assigned to the different nodes of the network by adopting adequate key predistribution mechanisms. The remaining of the paper is organized as follows. Section 2 presents existing work. Section 3 describes the proposed key management scheme, while in Section 4 a performance analysis and evaluation of the proposed scheme is provided based on simulations results. Section 5 describes the security analysis of the proposed scheme, and finally conclusions are provided in Section 6.

2. Related Work

To make a system secure, cryptography is considered as one of the important security blocks which helps in implementing many security features. However, the management of these cryptographic keys has always been a challenging task.

To secure wireless sensor networks, Perrig et al. [10] proposed SPINS, in which there is a secure central entity called server which is responsible for establishing a key among the sensor nodes. Before the network deployment, each node is assigned a secret key while its corresponding key is assigned to the base station. Each sensor node needs to authenticate itself to the server (base station) during the initialization phase using the assigned key. Since it is based on centralized base station approach, two sensor nodes can only establish a secret key through its centralized trusted base station. However, the failure of base station severely affects the performance of network because of the existence of single centralized entity (base station).

To overcome the abovementioned issue, a randomly key distributed approach is proposed by Eschenauer and Gligor [7]. In this scheme, there is no centralized entity like a base station for key distribution and management. Each node in the network is assigned a set of randomly selected keys from a large key set. Since the keys are distributed randomly, the two communicating nodes need to have at least one common key in their sets for secure communication. However, the nonexistence of a common key in their sets affects the network connectivity but it improves the network security against node capturing attacks. To further improve the network security, sharing of at least q-keys concept for establishing a secret key is introduced by Chan et al. [11]. This scheme improves the network security but it further degrades the network connectivity and increases the memory requirements. But the prior knowledge of node's deployment in the network helps in increasing the network connectivity and reduces the memory requirements [12] combined with Rabin's scheme [13]. The use of Rabin's scheme makes the approach computationally expensive. Hence to achieve better security and network connectivity with less memory requirements with low computational cost, NPKPS scheme is proposed by Zhang et al. [14] for wireless sensor networks.

The above described approaches are suitable for the static networks only. Because introducing node's mobility needs to increase the size of assigned key set which increases the memory requirements. Hence to reduce the memory cost, a level-based key management scheme for multicast communication is proposed by Kim and Ramakrishna [15] while a two-layered dynamic key management for clustered based wireless sensor networks is presented by Chuang et al. [16].

The management of secret keys (MASY) protocol is presented by Maerien et al. in [17] which is based on the trust assumption among the networks managers/base stations. When a node enters into an unknown networks, it can establish a secret key with the new network's manager/base station using the trust relationship assumption among the networks managers/base stations.

To further improve the network connectivity and reduce the memory requirements of the symmetric key distribution approaches, Du et al. [9] present an asymmetric key predistribution (AP) approach. Du sensor network model consists of two different types of nodes making it heterogeneous sensor networks (HSNs). This assumption significantly increases the network connectivity and reduces memory requirements compared to the existing symmetric key management approaches. Nodes with high capabilities act as cluster head and are assigned m keys, while nodes with low capabilities act as normal nodes and are assigned l keys, where $m \gg l$. This assumption increases the network resilience against node capturing attacks. Lu et al. [18] propose a framework for key management schemes in distributed peer-to-peer wireless sensor networks with heterogeneous sensor nodes and show by simulation that heterogeneity results in higher connectivity and higher resilience. Du et al. [19] propose a routing-driven key management scheme for heterogeneous wireless sensor networks, based on Elliptic Curve Cryptography (ECC), which provides better security with significant reduction of memory overhead. However, the prior knowledge of nodes deployment in the network further improves the network connectivity and resilience by reducing the memory cost [20].

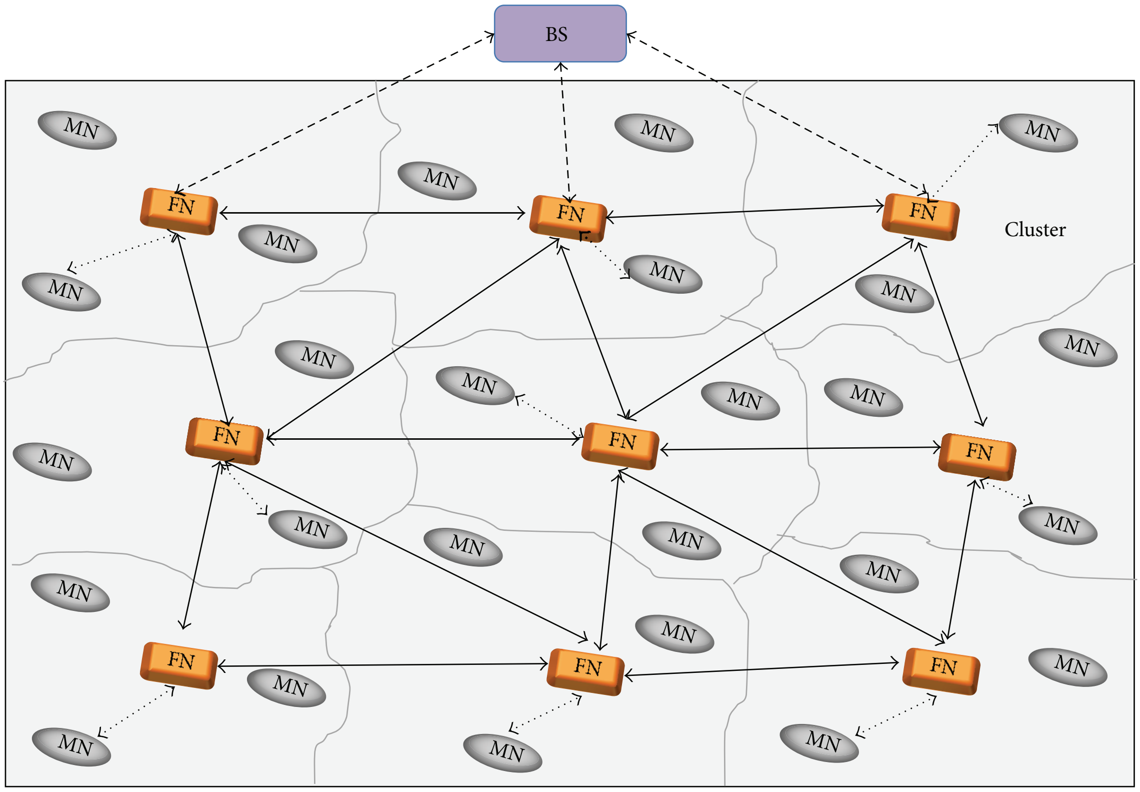


Figure 1: Virtual network architecture.

Network Model. The considered network model is a Heterogeneous Sensor Network (HSN) composed of H-sensors, that is, powerful nodes and L-sensors, that is, having limited capabilities. More specifically, H-sensors and L-sensors have different computational resources, available storage capacity, and battery lifetime, but we assume that all nodes use the same transmission power and have the same transmission rate. In addition, a Base Station (BS) acts as a reference sink node in HSNs and as a gateway offering connectivity towards other types of networks. With respect to mobility, H-sensors have the role of fixed nodes (FNs) thus defining, along with the BS, a sort of fixed WSN infrastructure, while L-sensors are considered as mobile nodes (MNs). The virtual network organization is shown in Figure 1. From the security point of view, sensor nodes are allowed to communicate only after being authenticated by the network and having established a communication key with the peer node.

In this proposed network architecture, the base station and the fixed nodes are more powerful than mobile nodes, hence given the responsibility to play an important role in the authentication and key management. More specifically, the fixed nodes act as cluster heads (CHs) while the mobile nodes act as cluster members. The selection of cluster head by the mobile node is based on the received signal strength of fixed nodes. Since the fixed nodes and mobile nodes are using the same transmission power, a MN may come in the coverage range of transmission of more than one fixed node. Hence a mobile node selects a fixed node with the highest signal strength as its cluster head.

To introduce the mobility in the network, we use a realistic mobility model in the OMNET++ simulator. The considered mobility model for the proposed scheme is the random way-point model [21]. This model ensures that all the targeted destinations are equiprobable. Each node in the network is given (1) its initial deployed location, (2) target location, (3) velocity, and (4) time duration for taking a random decision. Once the timer expires, the node randomly chooses next location as a target and keeps its velocity constant and starts its journey toward the new selected target location in the given area. When it arrives to the new location, it repeats the process again. However, the selection of a target location in a given scenario is based on a uniform distribution.

Here we describe a list of abbreviations used in the proposed solution:

CH: cluster head,

MN: mobile node,

FN: fixed node,

KP_{main}: main large key pool,

KP_{FN}: subkey pool for fixed nodes,

KP_{MN}: subkey pool for mobile nodes,

K_{plc}: public key,

K_{prt}: private key,

prand(): prime number generator,

AUTH: authentication code,

PRM: generated prime number,

SP_{MN}: scalar product of a mobile node,

SP_{FN}: scalar product of a fixed node,

SCC: secret communication code.

3. Proposed Scheme

The proposed key management scheme is built on top of the above network model to provide effective authentication and dynamic key establishment. The key material is generated at the BS. More specifically, a large key pool KP_{mail} is created and then divided into two subkey pools KP_{FN} and KP_{MN} such that $KP_{FN} \cap KP_{FN} = \emptyset$.

The key pool KP_{FN} is used by the FNs of the network while the key pool KP_{MN} is used by the MNs of the network for the secret key establishment. For authentication purposes, Elliptic Curve Cryptography (ECC) is used during the initialization phase for key generation. Three different phases have been taken into account:

(1) key predistribution to the different sensor nodes, that is, FNs and MNs;

(2) node authentication;

(3) communication key establishment among the nodes within the network.

Further details will be provided in the following subsections. Furthermore, Figure 2 presents a graphical description of all the operations foreseen in the proposed solution.

3.1. Key Predistribution. As already mentioned, in our proposed scheme, the key material is organized at the BS in a large key pool KP_{main} which is then randomly divided into key pool KP_{FN} and into key pool KP_{MN} such that $KP_{FN} \cap KP_{MN} = \emptyset$. Now each FN i is assigned a randomly selected key pool KP_{FN_i} from the key pool KP_{FN} where $KP_{FN_i} \ll KP_{FN}$ and contains $|KP_{FN_i}|$ keys while each MN j is assigned a randomly selected key pool KP_{MN_j} from the key pool KP_{MN} where $KP_{MN_j} \ll KP_{MN}$ and contains $|KP_{MN_j}|$ keys. Since these two key pools are disjoint, $KP_{FN_i} \cap KP_{MN_j} = \emptyset$. These assigned key pools will be used by the FNs and by the MNs for the establishment of a secret communication key using the assigned key generation algorithm.

Concerning the authentication key material, each FN and each MN are assigned an elliptic curve $E(a, b)$ over a finite Galois field $F(G)$ and a base point G along with a unique authentication code AUTH. Each FN and each MN are also assigned an ECC-based public/private key pair (K_{plc}, K_{prt}) and a prime number generator (prand()).

As previously described, FNs and the BS compose the fixed infrastructure of the overall heterogeneous sensor network; they are powerful devices and are responsible for the authentication and key management services offered to the MNs. In order to maintain the availability of these services and to avoid the full network being compromised by attackers, a higher level of security is thus required for FNs and the BS. As a consequence, the authentication of FNs to the network and the communication between the FNs and between a FN and the BS will be based on a standard ECC-based private/public key mechanism. Accordingly, each FN has its own private key and the public key of the BS and of all the other FNs of the network. At the same time, the BS has the public keys of all the FNs.

All the previously introduced key material is transferred to each node of the network by means of secure side channels. Then, after this predistribution phase, the specific key material assigned to each type of node of the network is as follows:

(i) the BS owns all the key material that needs to be predistributed (plus, as already described, the public key of each FN);

(ii) each FN i has been given $E(a, b)$, G, and $AUTH_i$ for authentication purposes and key pool KP_{FN_i} for communication key establishment;

(iii) each MN j has been given $E(a, b)$, G, and $AUTH_j$ for authentication purposes and KP_{MN_j} for communication key establishment.

3.2. Node Authentication. After the deployment and key predistribution phase, each FN of the network broadcasts periodic Hello messages. This mechanism enables each FN to fill a table with all neighboring MNs. These Hello messages include the FN ID and a random nonce signed by the FN's private key. Upon the reception of those Hello messages, each MN selects a FN as its cluster head (CH), for example, the one with the highest signal strength, after the verification of Hello message by using the FN public key. Since Hello message verification is a part of the authentication phase, at this point the authentication phase among the FNs and the MNs can start. To this aim, each MN_j authenticates the Hello message of the selected FN_i as a CH as follows. First MN_j uses the FN_i ID and generates a prime number PRM_{FN_i} using the prime number generator prand()

$$PRM_{FN_j} = \text{prand}\left(ID_{FN_i}\right). \quad (1)$$

After the generation of PRM_{FN_i}, the MN_j generates the public key of the FN_i using the scalar multiplication as

$$K_{plc} = \left(PRM_{FN_i} + ID_{FN_i}\right) \cdot G. \quad (2)$$

Then the MN_j can verify the Hello message signature. Successful verification of the Hello message signature authenticates the CH, that is, FN_i to the MN_j. The MN then calculates the scalar product of the assigned authentication code $AUTH_j$ and its private key K_{prt} as

$$SP_{MN_j} = \left(AUTH_j + ID_{MN_j}\right) \cdot K_{prt}. \quad (3)$$

Then the MN_j sends a joining request including its ID, SP_{MN_j}, and the nonce it had received from the CH back to its selected CH, all signed by its private key. After receiving the MN_j's joining request message, the FN_i first authenticates MN_j

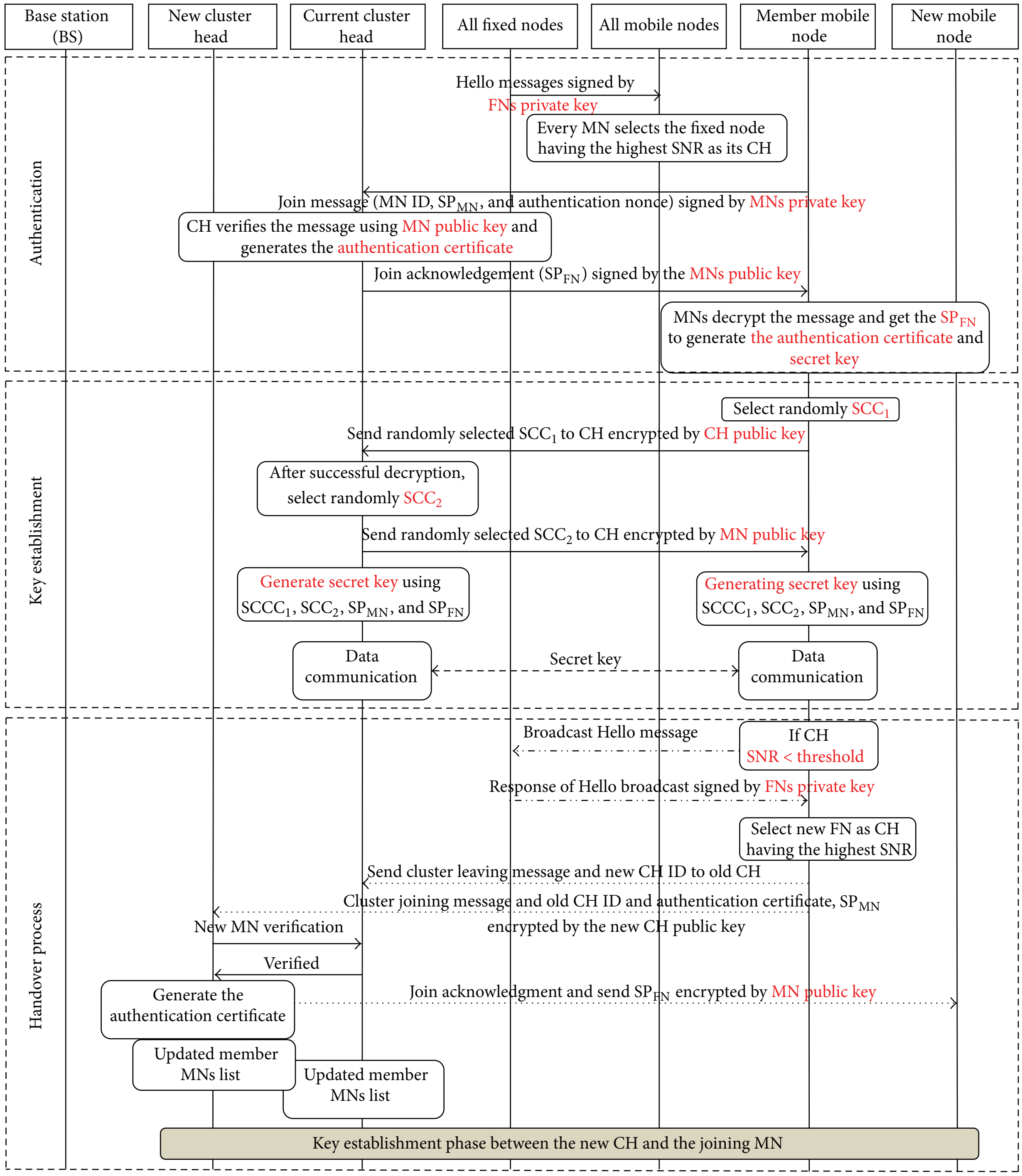

Base station (BS)
New cluster head
Current cluster head
All fixed nodes
All mobile nodes
Member mobile node
New mobile node
Authentication
Hello messages signed by FNs private key
Every MN selects the fixed node having the highest SNR as its CH
Join message (MN ID, SP_{MN}, and authentication nonce) signed by MNs private key
CH verifies the message using MN public key and generates the authentication certificate
Join acknowledgement (SP_{FN}) signed by the MNs public key
MNs decrypt the message and get the SP_{FN} to generate the authentication certificate and secret key
Key establishment
Select randomly SCC_1
Send randomly selected SCC_1 to CH encrypted by CH public key
After successful decryption, select randomly SCC_2
Send randomly selected SCC_2 to CH encrypted by MN public key
Generate secret key using $SCCC_1$, SCC_2, SP_{MN}, and SP_{FN}
Generating secret key using $SCCC_1$, SCC_2, SP_{MN}, and SP_{FN}
Data communication
Secret key
Data communication
Handover process
Broadcast Hello message
If CH SNR < threshold
Response of Hello broadcast signed by FNs private key
Select new FN as CH having the highest SNR
Send cluster leaving message and new CH ID to old CH
Cluster joining message and old CH ID and authentication certificate, SP_{MN} encrypted by the new CH public key
New MN verification
Verified
Generate the authentication certificate
Join acknowledgment and send SP_{FN} encrypted by MN public key
Updated member MNs list
Updated member MNs list
Key establishment phase between the new CH and the joining MN

Figure 2: Overview of the proposed key management scheme.

before registering it as a trusted cluster member. The FN_i follows the same procedure as the MN_j did to check the authenticity of the received messages. First the FN_i uses the MN_j ID and generates a prime number PRM_{MN_j} using the prime number generator prand()

$$\mathrm{PRM}_{\mathrm{MN}_j} = \mathrm{prand}\left(\mathrm{ID}_{\mathrm{MN}_j}\right). \qquad (4)$$

After the generation of PRM_{MN_j}, the FN_i generates the public key of the MN_j using scalar multiplication as

$$K_{\mathrm{plc}} = \left(\mathrm{PRM}_{\mathrm{MN}_j} + \mathrm{ID}_{\mathrm{MN}_j}\right) \cdot G. \qquad (5)$$

After the generation of the MN_j public key, the FN_i verifies the joining message signature. Successful verification and reception of the correct nonce ensure that the MN_j is an authentic mobile node belonging to the network. The CH registers this MN_j into its authentic MN member list and calculates the scalar product of AUTH_i and its private key as

$$\mathrm{SP}_{\mathrm{FN}_i} = \left(\mathrm{AUTH}_i + \mathrm{ID}_{\mathrm{FN}_i}\right) \cdot \mathrm{K}_{\mathrm{prt}}. \quad (6)$$

Finally the CH generates an authentication certificate for this MN using $\mathrm{SP}_{\mathrm{MN}_j}$ and $\mathrm{SP}_{\mathrm{FN}_i}$ as

$$\text{Authentication Certificate} = \mathrm{SP}_{\mathrm{MN}_j} \cdot \mathrm{SP}_{\mathrm{FN}_i} \bmod G. \quad (7)$$

The CH sends $\mathrm{SP}_{\mathrm{FN}_i}$ to the MN_j which is used in the secret key generation and for the authentication certificate generation.

3.3. Communication Key Establishment. Once the MN and CH/FN authenticate each other successfully, the key establishment phase starts. During this phase, the MN sends one of its secret communication codes SCC_1, randomly selected from $\mathrm{KP}_{\mathrm{MN}}$ and encrypted by the CH public key to its CH as described above. The CH also selects randomly another secret communication code SCC_2 from its pool $\mathrm{KP}_{\mathrm{FN}}$ and sends it to the corresponding MN. After the reception of this secret code by the MN, the MN and the FN both have the same SCC_1 and SCC_2. Now the MN and the FN use SCC_1, SCC_2, $\mathrm{SP}_{\mathrm{MN}_j}$ and $\mathrm{SP}_{\mathrm{FN}_i}$ to generate secret key using standard approach defined in [22] as

$$\text{Secret Key} = \mathrm{SCC}_1 \cdot \mathrm{SCC}_2 \bmod \left(\mathrm{SP}_{\mathrm{MN}_j} \cdot \mathrm{SP}_{\mathrm{FN}_i}\right). \quad (8)$$

Once a secret key is established between the CH and each MN, the CH has assigned a Shared Secret Code (SSC) to its all member MNs. This shared secret code is updated both periodically and when a MN compromission is detected. Since the MNs move in the network to perform their duties, they may need to establish a secure communication link also with neighboring MNs, possibly very frequently due to their movement within the network. In order to keep track of their neighboring MNs, each MN broadcasts a short range Hello message to know about its neighboring MNs. To establish a secret key with a neighboring MN, both MNs will share their secret communication code IDs assigned to them as $\mathrm{KP}_{\mathrm{MN}}$. Now both the MNs will find the maximum number of shared codes with one another and will generate a secret key using all of them as

$$\text{Secret Key} = \prod_{l=1}^{f} \mathrm{SCC}_{1l} \bmod \mathrm{SSC}, \quad (9)$$

where "f" represents the total number of common secret communication codes. Since the distributions of the SCC_1 codes to the MNs is random and probabilistic, two neighboring MNs might not have any secret communication code in common. In this case, to avoid any discontinuity, the MNs will use the assigned Shared Secret Code (SSC) from their common CH and their IDs to establish a secret key with its neighboring MNs. For example, if MN_m wants to establish a secret key with MN_n but these two nodes do not have any common secret communication code (SCC), then they establish a secret key by first calculating and sharing L and K with each other as

$$L = \mathrm{prand}\left(\mathrm{ID}_{\mathrm{MN}_n}\right) \cdot \mathrm{SP}_{\mathrm{MN}_m} \cdot \mathrm{AUTH}_m \cdot \mathrm{SSC} \bmod G, \quad (10)$$

$$K = \mathrm{prand}\left(\mathrm{ID}_{\mathrm{MN}_m}\right) \cdot \mathrm{SP}_{\mathrm{MN}_n} \cdot \mathrm{AUTH}_n \cdot \mathrm{SSC} \bmod G, \quad (11)$$

$$\text{Secret key} = L \cdot K \bmod \mathrm{SSC}. \quad (12)$$

3.4. Handover. The MNs are moving and they may leave their current CH and join a new CH. In order to know when they need to perform the handover to remain connected to the network, each MN monitors (e.g., by sending periodically a signal strength inquiry message) the SNR of its CH. Once the MN detects that its CH SNR is below a predefined threshold value, it broadcasts a request to find its neighboring FNs. Upon the reception of the response, the MN selects the FN with the highest SNR as its new CH. When the MN performs the transition from its old CH to the new CH, it sends its old CH identity, $\mathrm{SP}_{\mathrm{MN}_j}$, and authentication certificate to the new CH and a cluster leaving message to the old CH. The new CH communicates directly with the old CH of the MN, using its public key, to confirm the MNs transition and for the verification of the authentication certificate. Once the new CH receives the MNs transition confirmation and the authentication certificate verification, it adds this incoming MN to its trusted member MN list. Also the new CH generates its own $\mathrm{SP}_{\mathrm{FN}_i}$ and sends it to the MN to update the certificate.

3.5. Protection of Key Material. Since the sensor network may be deployed in unattended or even hostile areas, for example, when used for military applications, node capture and physical damage cannot be avoided. In order to protect the key material, we assume that each node is provided with tamper resistant hardware [23] and with a mechanism operating in such a way that when a node is captured and physically damaged, its keys become invalid. More specifically, once a manumission attempt is detected, both the original authentication and communication keys and the relevant IDs are replaced by fake ones: this would ensure that the original keys material is never revealed to an adversary.

Once the FN receives a joining request and key request from a compromised MN, it will inform the BS and all its neighboring FNs about the compromised MN's ID so that each FN can delete the key map of this compromised MN. In case the FN itself is compromised, however, it would not be able to generate the secret key and when it tries to authenticate the MN, the MN will understand that the FN is compromised. The MN would then contact other neighboring FNs for authentication and communication key establishment and eventually notify them about the compromised FN. This information will be also propagated to the BS for verification. The BS will then ask the compromised FN about its communication key IDs. If the BS receives the wrong key IDs, then the BS would declare this FN as compromised and inform all the other FNs of the network about it.

4. Performance Evaluation

In this section, we discuss the results obtained by using OMNET++ simulators. In order to introduce the mobility in OMNET++ simulation, we use MixiM 2.0.1 framework. Simulations are based on a network with 400 mobile nodes (MNs) and 16 fixed nodes (FNs). The area where the MNs are deployed and move is 400 m × 400 m. As mentioned earlier, we are using the same transmission power for both the fixed nodes and the mobile nodes, using the CSMA 802.15.4 standard and the radio specifications based on the CC2420 radio chip. The velocity of the mobile nodes is kept constant at 1 m/s while the next target's location selection interval is set to 0.1 s. The transmission power is set to 10 mW and the receiver sensitivity is set to −95 dBm.

4.1. Connectivity. Network connectivity is considered an essential part in evaluating the performance of the network. From the security point of view, two neighboring nodes in a network are said to be connected if they have a secret key for secure communication. In this section, we are evaluating the performance of the proposed scheme against some existing schemes in terms of probability that two nodes can authenticate each other and establish a secret key (i.e., connectivity). In the scheme discussed above, we use both (1) online key generation for authentication and (2) secret communication key establishment and key predistribution to ensure one-hop network connectivity among the MNs.

Here the network connectivity is evaluated by using the OMNET++ simulation results. In case of the key predistribution techniques, the probability of single key sharing defines the network connectivity and is given by

$$\Pr[\text{Match}] = \frac{(|\text{KP}_{\text{main}}| - |\text{KP}_{\text{MN}}|)!\,(|\text{KP}_{\text{main}}| - |\text{KP}_{\text{FN}}|)!}{|\text{KP}_{\text{main}}|!\,(|\text{KP}_{\text{main}}| - |\text{KP}_{\text{MN}}| - |\text{KP}_{\text{FN}}|)!}, \quad (13)$$

where KP_{main} is a large key set containing $|\text{KP}_{\text{main}}|$ keys from which a key set KP_{MN} containing $|\text{KP}_{\text{MN}}|$ keys is assigned to each mobile node and a key set KP_{FN} containing $|\text{KP}_{\text{FN}}|$ keys is assigned to each fixed node, where $|\text{KP}_{\text{FN}}| \gg |\text{KP}_{\text{MN}}|$. In balanced key distribution schemes where each node of a homogeneous sensor networks is assigned the same number of keys, that is, $\text{KP}_{\text{MN}} = \text{KP}_{\text{FN}} = \text{KP}_{\text{node}}$, and the single key sharing probability is given by

$$\Pr[\text{Match}] = \frac{(|\text{KP}_{\text{main}}| - |\text{KP}_{\text{node}}|)!^2}{|\text{KP}_{\text{main}}|!\,(|\text{KP}_{\text{main}}| - 2|\text{KP}_{\text{node}}|)!}. \quad (14)$$

Figure 3 represents the OMNET++ simulation results comparing the connectivity of a MN to the CH with some of the existing schemes [7, 9, 14, 24]. It is clear that the online authentication key generation technique substantially improves the network connectivity because the network connectivity provided by the proposed scheme is almost 100%.

In order to further analyze the effect of key pool size on network connectivity in the key predistribution schemes, we carried out OMNET++ simulation to find the key sharing probability between the FNs and MNs by limiting the MN key pool sizes ($|\text{KP}_{\text{MN}}|$) to 10, 20, and 30 and the FN key pool sizes ($|\text{KP}_{\text{FN}}|$) to 450, 550, and 650 from a large key pool of size $|\text{KP}_{\text{main}}| = 10,000$. A simulation of 10,000 seconds for each combination of FN and MN key pool size is performed. For each we evaluate how many key requests were received from MNs by FNs and for how many key requests the FN has a shared key. If the total number of key requests received by all the FNs is X and the total number of key requests for which FNs have a share key is Y, the key sharing probability is given by

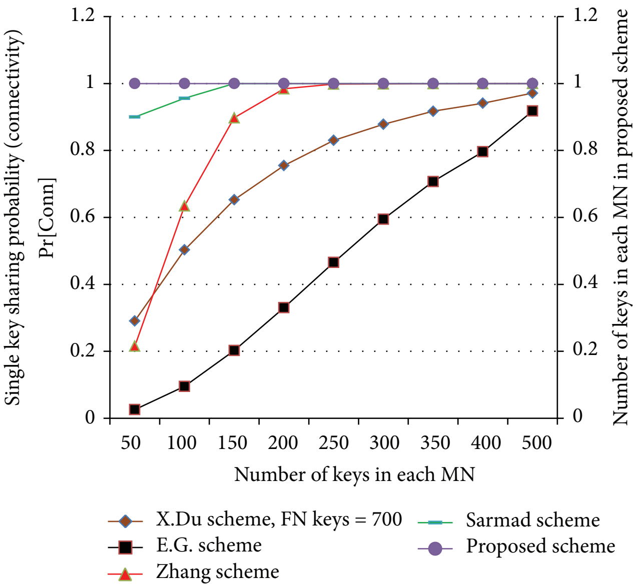


FIGURE 3: Single authentication key sharing probability between fixed node and mobile node.

$$\Pr[\text{Match}]_{\text{Simulation}} = \frac{X}{Y}. \quad (15)$$

Figure 4 represents the comparison of the effects of assigning different key pools to the MNs and FNs in terms of network connectivity obtained from OMNET++. The results show that the connectivity increases by assigning large key pools to the FNs and to the MNs in the existing key predistribution schemes while the proposed scheme is independent of the assigned key pool sizes and provides almost 100% network connectivity as shown in Figure 3.

The deployment of the fixed nodes is such that it covers the whole area of the network. Under this assumption, a mobile node may come into the coverage area of more than one fixed node; hence the probability of authentication of each MN by any in-range authentic FN is given by

$$\Pr[\text{Authentication}] = 1 - (1 - \Pr[\text{Match}])^d, \quad (16)$$

where d represents the total number of neighboring fixed nodes, Pr[Authentication] is the probability of authentication, and Pr[Match] is single key sharing probability defined by (11). Figure 5 shows that the coverage of each MN by more than one FN dramatically increases the authentication probability of the MN, which in turn improves the connectivity

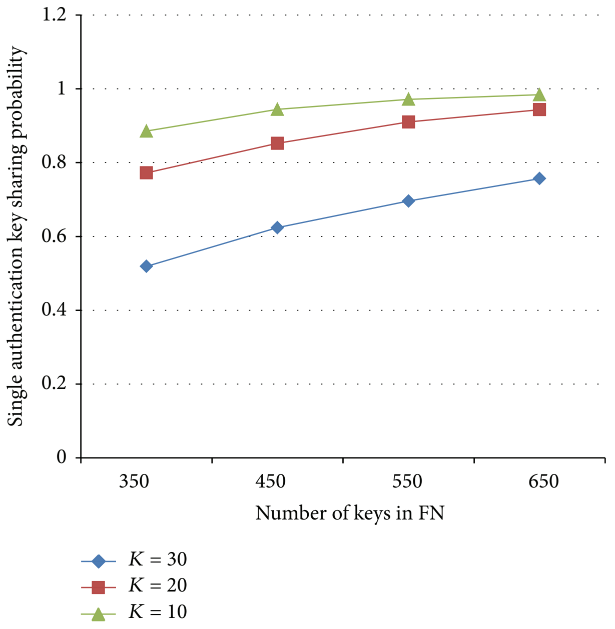


FIGURE 4: Single key sharing probability with different key pool size.

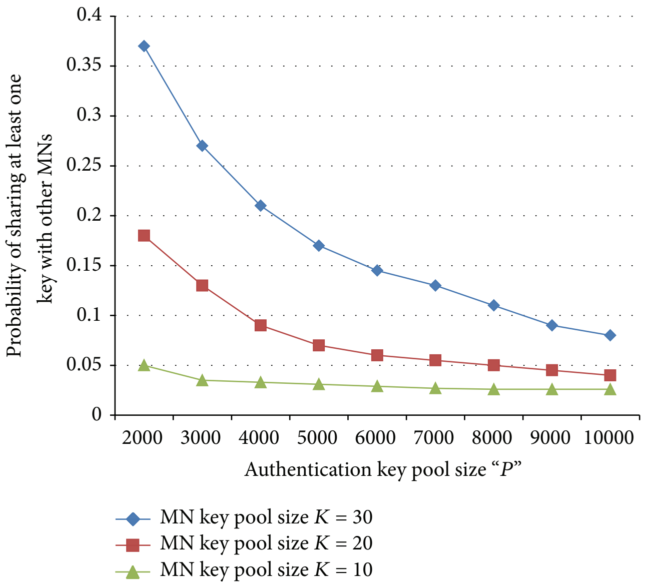


FIGURE 6: Single key sharing probability among the mobile nodes.

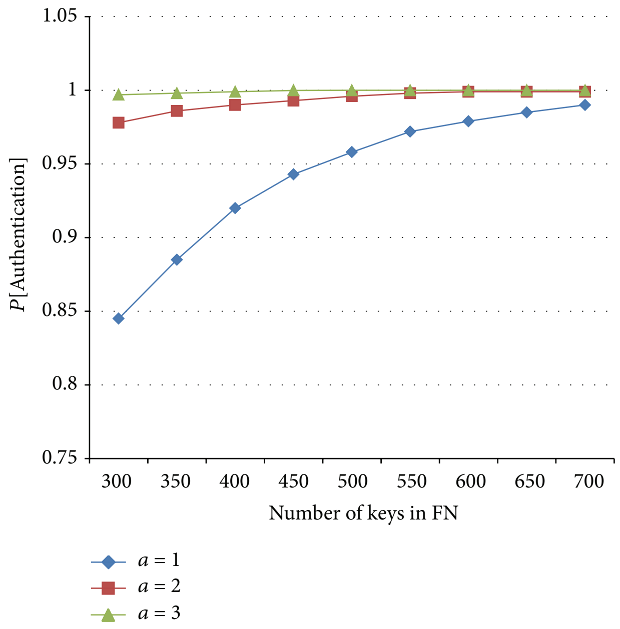


FIGURE 5: MN authentication probability by more than one FN $KP_{MN} = 30$.

of the MN to the network. However, it is still less than what is achieved by the proposed algorithm. This is because the selection of the FN depends on the link quality, availability, and bandwidth but the common key matching also plays an important role in case of key predistribution schemes, while it is not necessary in the proposed scheme. Figure 5 shows the OMNET++ result when the MN is within the range of one FN. So far, we assumed that each MN is at least under the radio coverage of one FN, but in a real network deployment, this may not be always possible. In that case, a MN may communicate with the FN through another intermediate MN. For that, a MN must share at least one common key with the other MNs in case of the key predistribution schemes, which further affects the performance of the network in terms of connectivity. The probability of sharing at least one common key between the MNs can be obtained analytically using (12). Figure 6 shows the results obtained by using OMNET++ simulator.

4.2. Memory Cost. The secret communication code key pool KP_{main} is divided into two subkey pools KP_{FN} for the FNs and KP_{MN} for the MNs such that $KP_{FN} \cap KP_{MN} = 0$. Each individual key SCC_1 of KP_{FN} can be used to create a secret key with all the keys of KP_{MN} and vice versa. Thus the total possible number of secret keys is

$$\text{Total secret keys} = \left|KP_{FN}\right| * \left|KP_{MN}\right|. \quad (17)$$

Let us assume that $|KP_{FN}| = |KP_{MN}| = 30$ keys. The total possible number of secret keys generated by these two key pools is 900. If we compare our proposed scheme with the existing schemes proposed by Eschenauer and Gligor in [7], instead of storing 900 keys in a sensor node, only 30 keys are required in each FN and in each MN to get 900 possible secret key combinations. This reduces the memory cost of the node while maintaining the same level of security by assigning only 30 keys instead of assigning 900 keys.

4.3. Node Energy Consumption. In this section, the energy consumption in the authentication and in key establishment phases has been observed using OMNET++ by optimizing the proposed algorithm in terms of the total number of exchanged message. Figure 2 shows that we need only two messages in the proposed scheme for the authentication purpose compared to [25–27] which require 4, 3, and 3 messages,

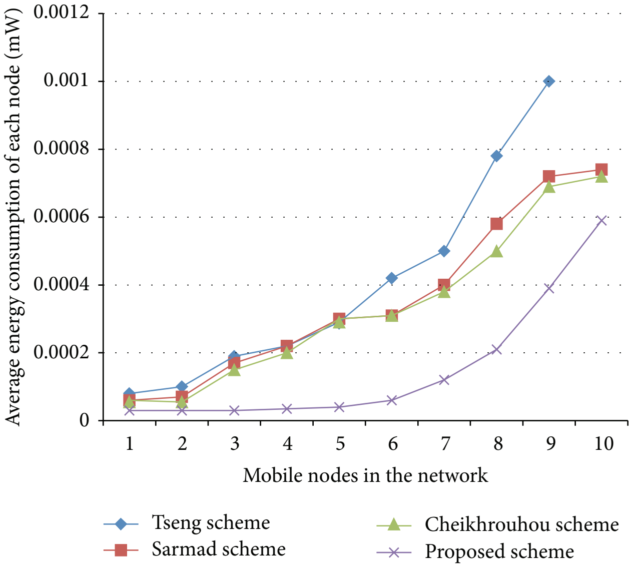


Figure 7: Overall energy consumption of a node.

respectively. Hence it is clear that less messages exchange consumes less energy in the proposed scheme. Again the total number of messages required in key establishment phase has been reduced to only two messages in the proposed scheme which is less than the existing approaches [25–27]. Hence the overall message exchanges by each node in the authentication phase and in the key establishment phase, including the acknowledgement, of the proposed algorithm are 5 while in the other schemes [25–27] they are 6, 7, and 6, respectively.

Figure 7 shows the comparison of the overall energy consumption of a node in the proposed algorithm with [25–27]. It is clear from the results that the proposed scheme consumes less energy than other algorithms. Hence it increases the network life time.

5. Security Evaluation

5.1. Denial of Service Attack. In this section we describe some kind of Denial of Service attacks (DoS attacks) that can be brought against our proposed scheme, as well as possible countermeasures. The main objective of DoS attacks is to make the resources unavailable to an intended user of the network.

(1) FN Hello Messages. The first possible DoS attack against the proposed scheme is to broadcast Hello messages pretending to be a FN of the network to exhaust the resources of the MNs. Since each Hello message is signed by the private key of the FN, MNs will verify it using the public key of that FN. Since the adversary FN is not an authentic node, the MN would not be able to verify that Hello message and once a MN detects this attack, it will inform its other neighboring authentic FNs. The authentic FNs would then inform the BS and neighboring MNs about this fake FN ID so that they can avoid the messages from that node.

(2) MN Hello Messages. When a MN finds its current CH signal strength value below a threshold value, it starts broadcasting the MN Hello messages to know about its new neighboring FNs. The attacker can launch such MN Hello message broadcast attack by introducing a fake MN. Since the MN Hello broadcast message is also signed by the MN private key, the new FNs first verify it by using the MN public key. This would not be possible for a fake MN. Thus the FNs inform the BS and other neighboring FNs about this malicious MN.

5.2. Sybil Attack. Sybil attacks are those in which a malicious node illegitimately takes on multiple identities. We call the nodes performing these attacks as Sybil nodes. Sybil attacks can be of different forms, for example, using direct or indirect communication and fabricated or stolen identities. In the direct communication Sybil attacks, a Sybil node communicates directly with a legitimate node. But since, in the proposed scheme, the Sybil node is first authenticated by sending a message signed with its private key, the FN would not be able to authenticate it. In the indirect communication Sybil attacks, malicious node (who deploys Sybil nodes in the network) becomes a router for forwarding the communication to the Sybil node from the FN which is not possible in the proposed scheme because each MN is the end user of the network. In the fabricated Sybil attacks, the attacker assigns an unuse identity to the Sybil node. In this case, this Sybil node needs to authenticate itself to the FNs which would again not be possible in the proposed scheme as described above. Stolen identity based Sybil attacks are very dangerous in such resource constrained networks. But this type of Sybil attack does not affect the proposed scheme because each communication is encrypted with the key agreed already with the original node having this ID, and the Sybil node does not have these keys.

However, we compare the proposed scheme with the balanced key predistribution scheme [7] and an unbalanced key predistribution scheme [9] to show the effectiveness of proposed scheme. Although the proposed scheme is partially based on key predistribution approach by assigning $|KP_{FN}|$ to the FNs and $|KP_{MN}|$ to the MNs but in the proposed scheme $|KP_{FN}| \cap |KP_{FN}| = \emptyset$ while in [7, 9], it should be $|KP_{FN}| \cap |KP_{FN}| \neq \emptyset$.

In the key predistribution approach, if every MN is assigned $|KP_{MN}|$ keys and every FN is assigned $|KP_{FN}|$ keys from a key pool of size $|KP_{main}|$ and an attacker compromises "c" nodes to create a compromised key pool of size "n", then the probability of a Sybil node to be successfully created is

$$\mathrm{Pr}_{\text{sybil node}} = \sum_{t=1}^{KP_{MN}} \frac{\binom{n}{t}\binom{KP_{main}-n}{KP_{MN}-t}}{\binom{KP_{main}}{KP_{MN}}} \frac{\binom{KP_{main}-KP_{MN}+t}{KP_{MN}}}{\binom{KP_{main}}{KP_{MN}}}. \qquad (18)$$

Figure 8 shows the probability of successfully generated Sybil nodes in the proposed scheme compared with scheme [7, 9].

5.3. Node Compromission. As described in Section 3.5, we assume that a node is secured by hardware means against

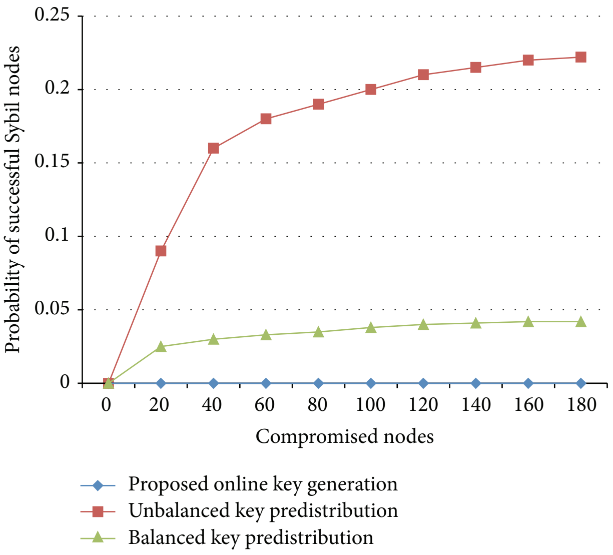


Figure 8: Probability of generation Sybil nodes.

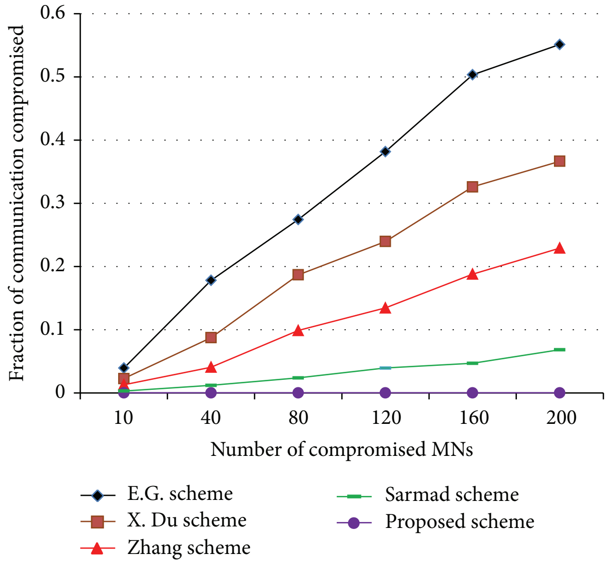


Figure 9: Network resilience against compromised mobile nodes.

access to its keys. However, no such scheme is ever perfect; hence here we analyze the effects of such attacks on our key management scheme.

In existing key predistribution schemes for both homogeneous and heterogeneous sensor networks, each node is assigned a key pool, and for secure communication the two nodes must have a shared common key. In that case, once the node is compromised by an adversary, it can compromise all the secure links with neighbors with whom this node has a shared key. Thus the fraction of communications compromised by compromising c MNs is given by

$$P\left[\text{Compromised}\right] = 1 - \left(1 - \frac{\left|\mathrm{KP}_{\mathrm{MN}_j}\right|}{\left|\mathrm{KP}_{\mathrm{MN}}\right|}\right)^{c}, \tag{19}$$

where $|\mathrm{KP}_{\mathrm{MN}_j}|$ is the number of keys stored in the MN and $|\mathrm{KP}_{\mathrm{MN}}|$ is the size of the authentication key pool from which $\mathrm{KP}_{\mathrm{MN}_j}$ is randomly selected for each MN. Figure 9 shows the OMNET++ simulation results of the effect of this kind of attack on our proposed scheme compared with the key predistribution scheme in [7, 9, 14, 24]. The graph shows that our scheme provides almost 100% resilience against this kind of attack.

6. Conclusion

In this paper, we proposed a new authentication and key management scheme for heterogeneous sensor networks including mobile nodes. The relevant network and mobility models have been presented as well. The proposed key management scheme is based on two different types of the key pools, that is, an authentication key pool and a communication key pool. Based on these pools, a key predistribution mechanism has been defined. Moreover, we compared our solution with some of the existing key management protocols for both homogeneous and heterogeneous sensor networks. The results showed that the two considered key pools not only provide better network connectivity in terms of authentication key sharing in the mobile scenario but also offer better security, while consuming less memory space compared with balanced key predistribution protocols. Furthermore, the proposed solution provides better network resilience against the node capture attacks compared to the other reference protocols considered.

Conflict of Interests

The authors declare that there is no conflict of interests regarding to the publication of this paper.

Acknowledgment

This paper has been partially supported by the European FP7 project BUTLER, under Contract no. 287901.

References

[1] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, “A survey of key management schemes in wireless sensor networks,” *Computer Communications*, vol. 30, no. 11-12, pp. 2314–2341, 2007.

[2] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[3] E. J. Duarte-Melo and M. Liu, “Data-gathering wireless sensor networks: organization and capacity,” *Computer Networks*, vol. 43, no. 4, pp. 519–537, 2003.

[4] K. Xu, X. Hong, and M. Gerla, “An ad hoc network with mobile backbones,” in *Proceedings of the IEEE International Conference on Communications (ICC ’02)*, vol. 5, pp. 3138–3143, May 2002.

[5] L. Girod, T. Stathopoulos, N. Ramanathan et al., “A system for simulation, emulation, and deployment of heterogeneous sensor networks,” in *Proceedings of the 2nd International Conference*

on Embedded Networked Sensor Systems (SenSys '04), pp. 201–213, 2004.

[6] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh, "Exploiting heterogeneity in sensor networks," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 2, pp. 878–890, March 2005.

[7] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communication Security*, vol. 47, pp. 41–47, November 2002.

[8] M. Rahman, N. Nasser, and K. Saleh, "Identity and pairing-based secure key management scheme for heterogeneous sensor networks," in *Proceedings of the 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communication (WiMob '08)*, pp. 423–428, October 2008.

[9] 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.

[10] 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.

[11] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 197–213, May 2003.

[12] F. Liu, J. Rivera, and X. Cheng, "Location-aware key establishment in wireless sensor networks," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '06)*, pp. 21–26, July 2006.

[13] M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," Tech. Rep. MIT/LCS/TR-212, Laboratory for Computer Science, MIT, 1979.

[14] J. Zhang, Y. Sun, and L. Liu, "NPKPS: a novel pairwise key pre-distribution scheme for wireless sensor networks," in *Proceedings of the IET Conference on Wireless, Mobile and Sensor Networks (CCWMSN '07)*, pp. 446–449, Shanghai, China, December 2007.

[15] K. T. Kim and R. S. Ramakrishna, "A level-based key management for both in-network processing and mobility in WSNs," in *Proceedings of the IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS '07)*, pp. 1–8, Pisa, Italy, October 2007.

[16] I.-H. Chuang, W.-T. Su, C.-Y. Wu, J.-P. Hsu, and Y.-H. Kuo, "Two-layered dynamic key management in mobile and long-lived cluster-based wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 4145–4150, IEEE, Kowloon, Hong Kong, March 2007.

[17] J. Maerien, S. Michiels, C. Huygens, and W. Joosen, "MASY: MAnagement of Secret keYs for federated mobile wireless sensor networks," in *Proceedings of the IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '10)*, pp. 121–128, Niagara Falls, Canada, October 2010.

[18] K. Lu, Y. Qian, and J. Hu, "A framework for distributed key management schemes in heterogeneous wireless sensor networks," in *Proceedings of the 25th IEEE International Performance, Computing, and Communications Conference (IPCCC '06)*, p. 520, Phoenix, Ariz, USA, April 2006.

[19] X. Du, S. Ci, Y. Xiao, M. Guizani, and H.-H. Chen, "A routing-driven key management scheme for heterogeneous sensor networks," in *Proceedings of the IEEE International Conference on Communications, (ICC '07)*, pp. 3407–3412, June 2007.

[20] Q. Yang, Q. Li, and S. Li, "An efficient key management scheme for heterogeneous sensor networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, pp. 1–4, Dalian, China, October 2008.

[21] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

[22] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[23] FIPS 140-2 Standard, "Security Requirements for Cryptographic modules," 2001.

[24] U. K. Sarmad, L. Lavagno, and C. Pastrone, "A key management scheme supporting node mobility in heterogeneous sensor networks," in *Proceedings of the 6th International Conference on Emerging Technologies (ICET '10)*, pp. 364–369, Islamabad, Pakistan, October 2010.

[25] 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, November 2007.

[26] O. Cheikhrouhou, A. Koubaa, M. Boujelben, and M. Abid, "A lightweight user authentication scheme for wireless sensor networks," in *Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications (AICCSA '10)*, pp. 1–7, Hammamet, Tunisie, May 2010.

[27] S. U. Khan, L. Lavagno, C. Pastrone, and M. Spirito, "An effective key management scheme for mobile heterogeneous sensor networks," in *Proceedings of the International Conference on Information Society (i-Society '11)*, pp. 98–103, June 2011.

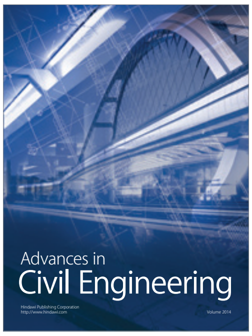
Journal of
Engineering
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014

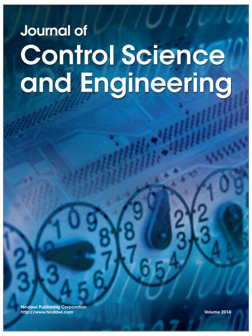
The Scientific World Journal
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014

International Journal of
Rotating Machinery
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014

Journal of
Sensors
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014

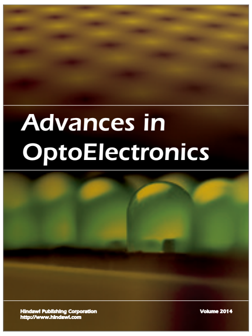
International Journal of
Distributed Sensor Networks
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014


Advances in
Civil Engineering
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014


Journal of
Control Science and Engineering
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014

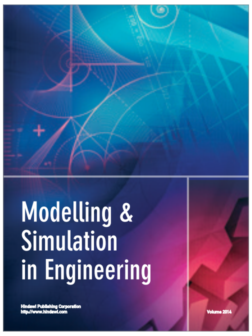
Journal of
Robotics
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014

Journal of
Electrical and Computer Engineering
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014

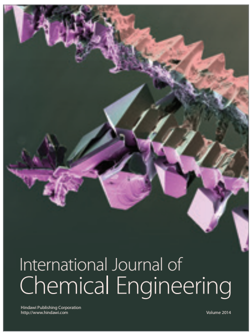

Advances in
OptoElectronics
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014

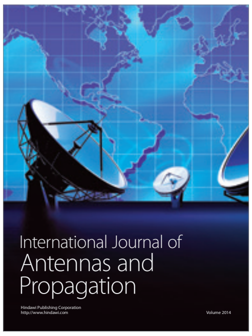
VLSI Design
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014

International Journal of
Navigation and Observation
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014

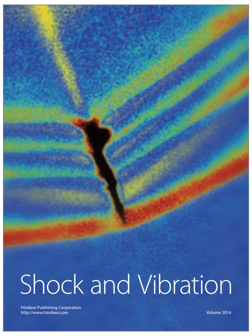

Modelling & Simulation in Engineering
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014

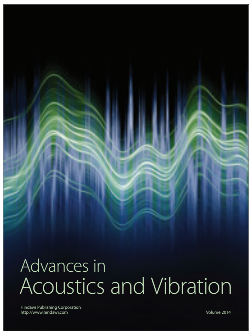
International Journal of
Aerospace Engineering
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014


International Journal of
Chemical Engineering
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014


International Journal of
Antennas and Propagation
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014

Active and Passive Electronic Components
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014


Shock and Vibration
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014


Advances in
Acoustics and Vibration
Hindawi Publishing Corporation
http://www.hindawi.com
Volume 2014