Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 303501, 9 pages
http://dx.doi.org/10.1155/2014/303501

Review Article

Security Analysis in Wireless Sensor Networks

Murat Dener

Graduate School of Natural and Applied Sciences, Gazi University, Besevler, 06500 Ankara, Turkey

Correspondence should be addressed to Murat Dener; muratdener@gazi.edu.tr

Received 15 May 2014; Revised 30 September 2014; Accepted 1 October 2014; Published 23 October 2014

Academic Editor: Lisimachos Kondi

In recent years, wireless sensor network (WSN) is employed in many application areas such as monitoring, tracking, and controlling. For many applications of WSN, security is an important requirement. However, security solutions in WSN differ from traditional networks due to resource limitation and computational constraints. This paper analyzes security solutions: TinySec, IEEE 802.15.4, SPINS, MiniSEC, LSec, LLSP, LISA, and LISP in WSN. The paper also presents characteristics, security requirements, attacks, encryption algorithms, and operation modes. This paper is considered to be useful for security designers in WSNs.

1. Introduction

Developments in low-cost sensor architectures have made wireless sensor networks (WSNs) a new and known research area [1]. These networks consist of large number of low-power and low-cost sensors with limited capacity, short-range transmitters spatially distributed in an often inaccessible and unreliable environment [2]. Each node has the abilities of calculation, detection, and communication [3]. These nodes that can be randomly distributed in the environment to be observed can recognize each other and can perform the task of measuring in a wide area by working together. Because of these properties, they can be used in a wide range of areas from health care to military, building security to detection of forest fires [4]. The WSN is facing a wide variety of security vulnerabilities due to the hardware limitations of the sensor nodes, wireless communication environment, real-time processing needs, heterogenic structure, large number of nodes, need for measurability, mobility, the weight of the application environmental conditions, and cost [5]. Confidentiality which is the basic goal of security provides one of the most important obstacles to overcome in order to ensure the integrity and availability as well as the achievement of time-critical and vital goals [6]. During sensitive WSN applications, such as the surveillance of enemy or borderlines, the security protocols which enable the sensors to transfer secret data to the base station must be used. However, the low processor and radio capacities of the sensors prevent traditional security protocols from being used in WSN applications [7]. Nowadays, various security protocols that consider these aspects of WSNs and their nodes are being developed. The security protocols to be developed should implement all the security issues (data confidentiality, data integrity, data freshness, data authentication, and availability) [8] but also provide high security with low energy consumption. Moreover, the fact that most of the suggested solutions are just based on the simulation platform and that solutions on sensory platforms are not considered is a big deficiency in past research. Thus, in order to be able to use the suggested protocols in applications that require solid security, the protocols should also be tested on sensor nodes besides the simulation platform. TinyOS is installed on the sensor nodes that compose the WSN. TinyOS is an embedded operating system distributed free of charge and with open source code to be used in wireless sensor networks. TinyOS is coded in NesC programming language. With this coding, the nodes can be imparted with new features. Designed algorithms or protocols can be installed on the nodes by using NesC programming language. TinyOS operating system is designed to support the needs of wireless sensor networks [9]. While trying to fulfill these requirements, it should not be forgotten that WSN has restricted energy sources and the primary goal of a WSN is energy efficiency. Otherwise, a protocol that fulfills all the security requirements but consumes a bit of too much energy will be just impractical for WSN. Therefore, to provide the security requirements and the security solutions, the methods they use and their variations in the literature must be very well known by the researchers developing a new security solution. In this study, security solutions in WSN are analyzed in detail. In the second chapter, WSN characteristics, security requirements, and attacks are given. In the third chapter, encryption algorithms and modes of operation are mentioned. While in the fourth chapter the current security protocols are described, analysis of the protocols is in the fifth chapter.

2. Wireless Sensor Networks

In this chapter, WSN characteristics, security requirements, and attacks are described.

2.1. Characteristics. Characteristics preventing the use of traditional security protocols in WSNs and only belonging to WSN are summarized below. Taking into account the mentioned characteristics during design and development of protocols increases the usability of them [6].

2.1.1. Large Scale. General applications of WSNs require geographical coverage of large areas [10]. Number of nodes in WSNs may exceed tens of thousands [11].

2.1.2. Limited Resources. Requirement that WSNs must be with low installation and operation cost necessitates that sensor nodes should have simple hardware. For this reason, operation and communication resources in WSNs are limited. For example, one of the generic sensor types, TelosB, has 16-bit 8 Mhz processor, 48 KB main memory, and 1024 KB flash memory. Every protocol must be designed taking into account limitations in processor capacity, memory and radio communication [10].

2.1.3. Redundancy. Because of node redundancy, each event is detected by the multiple sensor nodes on the network and therefore increases the amount of data to be transferred over it. In other words, redundancy increases the amount of data sent to the base station and decreases the life duration of the network [10]. To get rid of data redundancy data, clustering protocols are used.

2.1.4. Security. WSN applications, such as military systems and medical monitoring systems, are very sensitive in terms of security. Due to the limited resources of the sensor nodes, traditional security mechanisms cannot be used in WSNs. For this reason, the security mechanisms of WSNs should be designed considering limited resources and malicious sensors [10].

2.2. Security Requirements. The so far listed security requirements of WSN are data confidentiality, data integrity, data freshness, and data authentication and availability [12–15]. These requirements are briefly explained hereinafter.

2.2.1. Data Confidentiality. Data confidentiality in WSN impedes access of unauthorized people to obtain data which is one of the crucial requirements in sensitive WSN applications. A sensor node should not relay on the data derived from the environment to its neighbors. The data collected on the nodes can be very sensitive, particularly in military applications. Furthermore, in numerous applications, nodes have to transmit highly sensitive data (e.g., key distribution) to other sensor nodes by means of wireless transmission environment. Additionally, routing data must also be kept secret against malicious nodes because these nodes can exploit these data and reduce the performance of the network. Due to these issues, establishing a safe communication channel is vital for data transmission in WSNs. The standard approach for preserving data confidentiality is the encryption of the data with a secret key. Since they consume low energy, encryption algorithms that rely upon secret key substructure are used in WSNs.

2.2.2. Data Integrity. Data confidentiality can prevent taking hold of data by malicious nodes; however, it cannot stop data from being altered by unauthorized persons. Data integrity ensures that the message will not be altered during communication. A malignant node can cause the network to work improperly by disrupting the message. Furthermore, the messages might be disrupted during transmission without actual presence of a malicious node. Thus, it is essential to utilize message authentication codes or cyclic codes to ensure data integrity.

2.2.3. Data Authentication. Since WSNs use public wireless environment, they need authentication mechanisms to pick up messages and deceptive packets that come from malicious nodes. Authentication mechanisms aid a node in verifying the identity of a node that it is in contact with. If there is no authentication, a malicious node can behave as if it was a different node and might acquire some sensitive data and also hamper proper operation of other nodes. In case only two nodes are in contact, authentication can be achieved by symmetric key cryptography. Transmitter and receiver can compute the verification code of all the messages sent by a common hidden key.

2.2.4. Data Freshness. In WSN structures, sensors send measurement data related to environment in which they are present through specific time intervals and then what matters is the delivery of the measurement times. It is possible that an attacker can retransmit the copy of old measurement values. It is therefore important to check that the data is new. A counter can be added to the message packet or a random number can be used during encryption to maintain data freshness.

2.2.5. Availability. Availability denotes WSN's capability in sustaining its service continuity even during denial-of-service DoS attacks. One of the methods to hinder the service is DoS type of attack. This type of attack focuses on making the target system incapable of damaging any one and also using up of all the sources of that system by regular or consecutive attacks. From perspective of technical terms, there is no takeover, capture, or "hacking." What is done is pressurizing of the victim system to use its sources and make the system inoperable to serve. It is possible that DoS attacks take place at any protocol layer of WSN and the selected victim might make the nodes inoperative. Besides the DoS attacks, excessive communication or calculation load might run out of the battery of the node faster than expected. Highly serious consequences might result from not providing availability to WSN. Let us take a military based application as an example, if some nodes do not function properly, then the enemy alliances might leak from these nonfunctional parts of WSN. Developing a detection and defense unit is essential to provide availability.

2.3. Attacks. Comparable to any wireless network, WSNs are suffering from many different attacks. In this section, we introduce the major attacks to WSNs.

2.3.1. Physical Layer

Jamming. One of the attacks interfering with the radio frequencies that a network's nodes are using is jamming [16, 17]. Typical defenses against jamming include variations of spread-spectrum communication such as frequency hopping and code spreading [17].

Tampering. Tampering is another type of physical layer attack. If a physical access is given to a node, an attacker can draw sensitive information such as cryptographic keys or other data on the node. The node may also be altered or replaced to create a compromised node controlled by the attacker. Tamper-proofing the node's physical package is one of the defenses to this attack [17].

2.3.2. Data Link Layer

Collision. A collision occurs when two nodes attempt to transmit on the same frequency simultaneously. A typical defense against collisions is the use of error-correcting codes [17].

Exhaustion. Repetitive collisions can also be made use of by an attacker to cause resource depletion. A feasible solution is to impose rate limits to the MAC admission control such that the network can disregard excessive requests, thus preventing the energy drain resulting from repeated transmissions [17].

Unfairness. Rather than blocking access to a service outright, an attacker can degrade it for gaining an advantage such as causing other nodes in a real-time MAC protocol to miss their transmission deadline. Using small frames reduces the effect of such attacks by decreasing the amount of time with which an attacker can take hold of the communication channel.

2.3.3. Network Layer

Selective Forwarding. A malicious node attempts to block the packets in the network by rejecting to forward or drop the messages passing through them. In addition, the malicious node may send the messages to the wrong path so that it can create unfaithful routing information in the network [18]. Using multiple paths to send data is one defense against selective forwarding attacks whereas the second defense is to detect the malicious node or presume that it has failed and looked for a different route [19].

Sinkhole Attack. The intent of the adversary is to attract almost all the traffic from a certain area by means of a compromised node, creating a metaphorical sinkhole with the enemy at the center. Sinkhole attacks typically work by making a compromised node appear particularly attractive to neighboring nodes in terms of routing algorithm [20]. This type of attack causes selective forwarding to be very simple because all traffic from a large area in the network will flow through the adversary's node.

Sybil Attacks. A single node duplicates itself and is presented in more than one location. The Sybil attack aims at fault tolerant schemes, for example, distributed storage, multipath routing, and topology maintenance. In a Sybil attack, a single node exhibits multiple identities to other nodes in the network. Authentication and encryption techniques can hinder an outsider from starting a Sybil attack on the sensor network [21].

Wormholes Attacks. In a wormhole attack, an attacker gets packets at one point in the network, "tunnels" them to another point in the network, and then replays them into the network from that point [22].

HELLO Flood Attacks. A large number of protocols utilizing HELLO packets naively assume that receiving such packets means that the sender is within the radio range and is therefore a neighbor. An attacker may use a high-powered transmitter to deceive a large area of nodes into believing they are neighbors of that transmitting node. Cryptography is mainly the current solution to these types of attacks [23].

2.3.4. Transport Layer

Flooding. An attacker may make new connection requests over and over until the resources required by each connection are depleted or reached a maximum limit [24]. Solution of this problem is to require each connecting client to evidence its dedication to the connection by solving a puzzle.

Desynchronization. The adversary repetitively pushes messages which convey sequence numbers to one or both of the endpoints. Requiring authentication of all packets communicated between hosts is one of the possible solutions to this type of attack [24].

3. Encryption Algorithms and Operation Modes

In this chapter, encryption algorithms to ensure the data confidentiality in WSNs and modes of operation are described.

3.1. Encryption Algorithms. Secure encryption is divided into two types as symmetric cryptography and asymmetric cryptography. While in asymmetric cryptography encryption and

Table 1: Comparison of encryption algorithms.

Algorithm name	Key size (bit)	Block size (bit)	Round
DES	56	64	16
3DES	168, 112, 56	64	48
DES-X	184	64	16
Blowfish	32–448, 8–128	64	16
Twofish	128, 192, 256	128	16
TEA, XTEA	128	64	64
XXTEA	128	64	It depends on the block size
AES	128, 192, 256	128	It depends on the key size
Skipjack	80	64	32
HIGHT	128	64	32

decryption processes are done by different keys, in symmetric cryptography, encryption and decryption are done by the same key. Although public key encryption is more robust and provides better security than secret-key encryption, it is not used in WSNs directly because of its slow performance and requirement of more memory. Symmetric cryptography algorithms are discussed mainly in two classes as block and bit stream encryption algorithms. Block encryption algorithms take fixed-length blocks of data to be encrypted into the encryption function and generate encrypted data blocks with the same length. As an example for these algorithms, AES, DES, Skipjack, RC5, and so forth can be given. However, bit-stream encryption algorithms take data as a streaming series of bits. In these Vernam-type algorithms, the random bit stream generation must not be in a self-repeating structure. Example algorithms are RC2, RC4, and so forth.

There are a number of widely used symmetric algorithms, which are listed and briefly described and analyzed as follows. Also, Comparison of encryption algorithms is given in Table 1.

3.1.1. Data Encryption Standard (DES)/3DES/DES-X. DES is a block cipher, one form of symmetric cryptography algorithms, which was devised by IBM and selected by the National Bureau of Standards (NBS) in the early 70s. Almost for over 25 years, it has been the standard encryption algorithm for civilian applications. It has been considered completely to be insecure because it has a short key length. Triple DES (3DES) is deemed to be temporarily secure enough and still has a wide usage. DES-X is another variant on the DES block cipher which is intended to enhance the complexity of a brute force attack utilizing a technique that is referred to as key whitening. Another reason for DES-X is that the speed of 3DES is unallowable in many cases. Thus, there is a need for an efficient way to fortify the DES [25].

3.1.2. Blowfish/Twofish. Blowfish was designed by Schneier in 1994 [26]. Since there is no effective cryptanalysis found, Blowfish is still considered to be secure. In addition, it provides a proper encryption performance in software implementation. However, Bruce Schneier himself recommended using a more advanced version, Twofish instead. Twofish is another block cipher published in 1998 by Counterpane Labs. One of the five advanced encryption standard (AES) finalists was Twofish. However, it was not chosen by NIST as AES because the winner of AES (Rijndael) was considered to have better performance than other finalists in both hardware and software in average. Twofish allows a wide range of tradeoffs between the size and speed. It is also designed to be efficient on a wide range of platforms. Even though it was not selected as AES, it may still be a suitable choice in our case due to the different platform.

3.1.3. Tiny Encryption Algorithm (TEA)/XTEA/XXTEA. The TEA is a block cipher presented in 1994 [27]. Minimizing the memory footprint and maximizing the speed is the aim of TEA. It is a Feistel type cipher that utilizes operations from mixed (orthogonal) algebraic groups. There are two variants of TEA—extended TEA (XTEA) and corrected block TEA (XXTEA), which were designed to correct weaknesses in the original TEA.

3.1.4. Rijndael Algorithm (AES). The winner of AES selected by NIST in 2000 was Rijndael. Substitution permutation network is a design principle that Rijndael is based on. It is fast in both software and hardware. Different from its predecessor DES, Rijndael does not use a Feistel network.

3.1.5. Skipjack Algorithm. Skipjack was developed by the U.S National Security Agency (NSA). It is one of the simplest and fastest block cipher algorithms, which is critical to embedded systems. Skipjack or a variant of Skipjack is now used in TinySec, SenSec, and MiniSec in wireless sensor networks [28–30].

3.1.6. Scalable Encryption Algorithm (SEA). Designed for processors with a limited instruction set, the scalable encryption algorithm was proposed by Standaert et al. The proposed design is parametric in the text, key, and processor size and provably secure against linear/differential cryptanalysis, allowing efficient combination of encryption/decryption and “on-the-fly” key derivation. Target applications for such routines include any context requiring low-cost encryption and/or authentication [31].

Table 2: Description of operation modes.

Mode	Description	Typical application
Electronic codebook (ECB)	By using the same key, each block of 64 plaintext bits is encoded independently.	Secure transmission of single values
Cipher block chaining (CBC)	The input to the encryption algorithm is the XOR of the next 64 bits of plaintext and the preceding 64 bits of ciphertext.	General-purpose block-oriented transmission Authentication
Cipher feedback (CFB)	Input is processed j bits at a time. Preceding ciphertext is used as input to the encryption algorithm to generate pseudorandom output, which is XORed with plaintext to create next unit of ciphertext.	General-purpose stream-oriented transmission Authentication
Outback feedback (OFB)	Like CFB, except that the input to the encryption algorithm is the preceding DES output.	Stream-oriented transmission over noisy channel
Counter (CTR)	Each block of plaintext is XORed with an encrypted counter. The counter is increased for each subsequent block.	General-purpose block-oriented transmission Useful for high-speed requirements
Output codebook block (OCB)	Each block of plaintext is XORed with a NONCE and L values. Tag value is produced for privacy.	Authentication Privacy

3.1.7. HIGHT Algorithm. HIGHT is another block cipher proposed by Hong, allowing low-resource hardware implementation, which is suitable for ubiquitous computing devices, for example, a sensor in wireless sensor network (WSN) or a RFID tag. HIGHT does not only perform simple operations to be ultralight but also contains sufficient security as a good encryption algorithm [32].

3.2. Operation Modes. Together with the selection of the correct encryption algorithm to ensure data confidentiality, the selection of operation mode is also important. Operating modes in cryptography are methods allowing safe repetitive usage of a block password under a single key. Data must be divided into separate parts in order to process variable length messages. The last part should be extended with a completion scheme accordingly to fit the block length of the password. An operation mode defines the way of encryption of every one of these blocks and, for this purpose, usually it uses a randomly generated extra value named initialization vector (IV).

Six commonly used modes of operation are defined by NIST, which are listed in Table 2 [33, 34].

Operation modes are created specifically to be used in encryption and identity authentication. Historically, operating modes are studied extensively under a variety of data exchange scenarios in terms of error propagation. Integrity protection emerged for an entirely different cryptographic purpose other than encryption. Some modern operation modes like OCB integrated encryption and identity authentication efficiently.

4. Security Protocols

In this chapter, TinySec, MiniSec, IEEE 802.15.4, SPINS, Lsec, LLSP, LISA, and LISP are described.

4.1. TinySec. TinySec [29] developed by the University of Berkeley is a link layer security architecture that has been included in the TinyOS version. Its design is based on ease of use and minimal load brought on sensor network. TinySec supports two different security options: encryption with identity authentication and only authentication. In identity authentication encryption, data is encrypted and an identity authentication code (MAC) is added to the package. However, in only authentication method, data is not encrypted but only authentication of the package is realized with a MAC. As it is understood from this, in TinySec, the identity authentication is a must for each package but encrypting the data is an option that can be decided according to the application. In encryption of messages, Skipjack block encryption, 8-bit initialization vector (IV), and code block chaining (CBC) are used. There is no restriction on keying method; in practice, a single key pair (one for the encryption of data and the other for the calculation of MACs) is selected for the whole network according to the desired level of security. TinySec at the tightest security level where identity authentication encryption is used brings 10% extra load on energy, delay, and band width. However, in cases where only authentication is used, this ratio drops to 3%.

4.2. SPINS. SPINS [35], developed by Berkeley University, consists of μTESLA protocol used in identity authentication broadcasting, SNEP protocol providing confidentiality, identity authentication between two nodes and data freshness, and a routing protocol based on these. SNEP offers the below possibilities:

(i) semantic security: semantic security, meaning an attacker listening to the network cannot obtain any information about the plain text even if more than one encrypted copy of the same plain text is received, is realized by a counter shared between the receiver and the sender and incremented in each message exchange;

(ii) identity authentication: the receiving node verifies the identity of the sender with the MAC used;

(iii) recursion protection: the counter in MAC prevents old messages to be sent again;

(iv) weak freshness: the counter used between the receiver and sender for semantic security ensures the message received is sent after the previous one;

(v) low communication overhead: keeping the counter on receiver and sender, not placing it in the message, reduces communication overhead.

In conventional approaches, identity authentication is done by asymmetrical methods. However, hardware restrictions of sensors are highly insufficient for the quite expensive asymmetrical methods. μTESLA gives the logic of asymmetry to identity authentication with symmetric methods. The sender creates a MAC for the message packages to be broadcasted by using a key known by only itself and by using a one-way function. It broadcasts the key of the message a certain time after the message is sent. Thus, the possibility of changing the contents of the package is removed. At the receiver end, the package kept in a buffer memory is authenticated by using this key. RC5 is used in encryption. For all this identity authentication process, μTESLA needs synchronization between the receiver and the sender even if it is loose.

4.3. LISP. LISP aims security solutions in large-scale wireless networks consisting of a large number of nodes with limited resources. To scale networks consisting of a large number of nodes Park and Shin divide them into clusters, select a head for each cluster, and create a key server. LISP [36] (lightweight security protocol) has a new switching mechanism. It uses switching mechanism by using head cluster and key servers. Below are the advantages of this method:

(i) it uses an effective key broadcast which do not need ACKs to be sent;

(ii) it uses check bits created without adding them to the data message;

(iii) it might recover the lost keys;

(iv) it refreshes key without data encryption or decryption.

The benefits of LISP in protecting critical information against attacks can be summarized as follows.

(i) Data integrity prevents tampering of data that is sent.

(ii) Access control is achieved by controlling the inputs to the network.

(iii) Key refreshing provides protection against nodes that may jeopardize the network.

LISP protocol may combine together with security the other services (routing, data distribution, and location). LISP is a flexible and energy-sensitive protocol. In addition, because it does not need ACK and other control packages, it is quite strong against DoS [37] attacks.

4.4. IEEE 802.15.4. IEEE 802.15.4 [38, 39] defines medium access and physical layers for wireless private area networks (WPANs). Although this protocol was not developed for WSN, it is used in WSNs because of its low power consumption, low cost, and flexibility. Currently, this protocol works on Micaz, TelosB nodes produced by the company CrossBow. ZigBee strong encryption AES-128 is used. Zigbee provides freshness. Controlling freshness prevents repeated attacks. Counter is reset when a new key is created. Zigbee provides integrity and prevents an attacker from changing the message. Integrity options are 0, 32, 64, and 128 bit, by default 64 bit. Zigbee provides authentication. Authentication tests whether the right person is reached or not and prevents the attacker showing the device like another one. Authentication is possible at the network and device levels. Authentication at the network level is achieved by using a public network key. Authentication at device level is achieved by using the unique link key between devices. Zigbee provides encryption and prevents an attacker from intercepting and listening. Zigbee uses 128-bit AES encryption. Encryption security is provided at the network and at the device level A public key used at the network level encryption. It prevents attacks because of very low memory usage. Device level encryption uses a common link key. Zigbee uses three types of keys. Master key provides long term security between two devices. Link key provides security between two devices. Network key provides security on the network.

4.5. LSec. LSec [40] provides authentication and authorization with simple key exchange scheme. Furthermore, it has protection mechanisms against data confidentiality, breaches, and illegal events. There is variety of security attacks on sensor networks. As examples of DoS, eavesdropping, replay attacks, tempering the message, and malicious nodes can be mentioned. To defend against these types of attacks, LSec uses data confidentiality, identity authentication, data integrity, defense against intruders, and some security mechanisms. These problems can be solved partially when the communication among the nodes is encrypted but a complete solution requires a strong key exchange and distribution scheme. LSEc provides identity authentication and authorization, simple secure key exchange, defense mechanism against breaches, data privacy, and usage of asymmetrical and symmetrical encryption together. LSec protocol is simulated on sensor network simulator and emulator (SENSE). There is no application of it.

4.6. LISA. LISA [41] includes security solutions listed below:

(i) semantic security: the same data is encrypted in different ways by increasing the value of the counter after each data;

(ii) identity authentication: it ensures that the data is from the right node;

(iii) protection against replay attacks: it prevents old messages from being repeated;

(iv) weak freshness: base station verifies that the message generated is after the previous one.

Table 3: Security requirements/protocols.

Security requirements/protocols	TinySec	SPINS	MiniSEC	LSec	LLSP	LISA	IEEE 802.15.4	LISP
Data confidentiality	+	+	+	+	+	+	+	+
Data integrity	+	+	−	−	+	+	+	+
Data authentication	+	+	+	+	+	+	+	+
Data freshness	−	+	+	−	+	+	+	−
Data availability	−	−	−	−	−	−	−	+
Implementation	TinyOS (Mica2)	−	TinyOS (TelosB)	−	−	−	TinyOS (MicaZ, TelosB)	−

4.7. MiniSec. MiniSec [42] is implemented on Telos platform. While TinySec provides low security at low power consumption, ZigBee [43] provides high security at high power consumption. According to the authors, MiniSec provides high security at low power consumption. Three techniques are used to achieve this. First, block encryption method is used to provide privacy and authentication. But there is only one pass over the data. Second, initialization vector (or IV) used as a very few bits. Third, basic gaps are used during unicast and broadcast communication. In the unicast mode, the power consumption of radio is reduced by making extra computations and using synchronized counters. In the broadcast mode, bloom filter mechanism is used. SkipJack is used as the encryption algorithm and OCB as the encryption mode. It is defenseless against DoS attacks.

4.8. LLSP. LLSP [44] provides minimum cost identity authentication, data integrity, and semantic security by using only symmetric security algorithms. The key mechanism determines key management issues in WSNs. It includes the questions of how the cryptograph keys are distributed, shared, and updated. An appropriate keying mechanism depends on the factors such as the target hazard model, the network communication in practice, security requirements, and ease of use. Keying mechanism is not discussed in the paper.

5. Recommendation and Discussion

General evaluation is seen in Table 3. Within the solutions in the literature TinySec, MiniSec, SPINS, Lsec, LLSP, LISA, LISP, only TinySec, and MiniSec have been implemented on sensor nodes. Despite the fact that it has been developed for wireless private networks, IEEE 802.15.4 has been used in WSN due to its low energy consumption, low cost, and flexibility. Other security protocols have not been implemented on sensor nodes. In order to guarantee data confidentiality, TinySec and MiniSec have used the Skipjack algorithm of 80-bit size. Yet, past research has shown that, for data confidentiality, the key size should be at least 128 bits. TinySec cannot prevent message retransmission attacks while MiniSec cannot guarantee data integrity. Also, this protocol cannot provide availability criteria. The nonfulfillment of availability criteria means that the specified procedure will be unguarded against DoS attacks. Although Lisp protocol provides to availability criteria, it is not implemented on sensor nodes.

6. Conclusion

When developing a security approach, the capacities of resources (memory, processor, and power supply) of wireless sensor nodes should be taken into consideration. It is an expected result that additional encryption mechanisms to increase security in WSN applications increase the node power consumption amounts and the average end-to-end delay times. Here it is important to determine the requirements of the application very well. In a simple large-scale or industrial WSN application, security is not so important, whereas power consumption is very significant. On the other hand, security is very crucial in military and health care applications while power consumption can be relatively ignored. For this reason, it is important to select an encryption algorithm and an encryption mode appropriate for the security solutions developed to be used in military and health care applications. According to experimental results, the encryption algorithms using 64-bit keys for data privacy can be broken in 3.5 months with super computers which can try 10^{12} passwords in a second. This time value is 5.4×10^{18} years for the ones using 128-bit encryption algorithms [45]. Here, although it may seem reasonable to use 128-bit encryption algorithm, it may not be correct to use it because the memory required for this algorithm or the password encryption/decryption time will be more. Nevertheless, the operation mode (CBC, OBC, etc.) required for the encryption algorithm to be used is also important. It means that even to provide only data privacy, it should be considered which algorithm and which operating mode to be used in detail. The security solution developed should be modular. It means that if the new encryption algorithms and modes appearing in the literature are better in terms of security, power consumption, memory usage, and delay issues, they must be able to be integrated into the developed security solution directly. It is needed to develop a security solution which conforms with every aspect of the security requirements (data privacy, data integrity, data freshness, identity authentication, and availability) of WSN, but by taking into account the idea of high security and low power consumption for each requirement. Also, it is a drawback for researches that most of the recommended security solutions remained in the simulation environment, and they are not tried on sensor platforms. For this reason, it is necessary that, for the recommended protocols to be used directly in applications requiring security, they should not only remain in the simulation environments but also applied on sensor nodes. When developing a security solution, the most appropriate one must be selected by taking into consideration the WSN characteristics, the security requirements, the attacks, and the current encryption algorithms and modes. However, the strategies applying the security protocols in the literature may also help the researchers. It is expected that this study help guiding the people working on the security issues in WSNs.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

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. 102–105, 2002.

[2] S. Özdemir, "Secure data aggregation in wireless sensor networks via homomorphic encryption," *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 23, no. 2, pp. 365–373, 2008.

[3] 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.

[4] M. Çakiroğlu and A. T. Özcerit, "Denial of service attack resistant MAC protocol design for wireless sensor networks," *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 22, no. 4, pp. 697–707, 2007.

[5] T. Kavitha and D. Sridharan, "Security vulnerabilities in wireless sensor networks: a survey," *Journal of Information Assurance and Security*, vol. 5, pp. 31–44, 2010.

[6] H. Chan and A. Perrig, "Security and privacy in sensor networks," *Computer*, vol. 36, no. 10, pp. 103–105, 2003.

[7] H. Çam, S. Özdemir, P. Nair, D. Muthuavinashiappan, and H. Ozgur Sanli, "Energy-efficient secure pattern based data aggregation for wireless sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 446–455, 2006.

[8] N. Bandirmali and I. Erturk, "WSNSec: a scalable data link layer security protocol for WSNs," *Ad Hoc Networks*, vol. 10, no. 1, pp. 37–45, 2012.

[9] S. Erboral, *Kablosuz Duyarga Ağlarında Veri Birleştirilmesi Ve Değerlendirilmesi*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, 2008.

[10] M. Meghdadi, S. Özdemir, and İ. Güler, "Security in wireless sensor networks: problems and solutions," *International Journal of Information Technologies*, vol. 1, pp. 35–40, 2008.

[11] 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, Inc., Glenwood, Md, USA, 2000.

[12] H. K. D. Sarma and A. Kar, "Security threats in wireless sensor networks," in *Proceedings of the 40th Annual IEEE International Carnahan Conference on Security Technology (ICCST '06)*, pp. 243–251, IEEE, October 2006.

[13] D. R. Raymond and S. F. Midkiff, "Denial-of-service in wireless sensor networks: attacks and defenses," *IEEE Pervasive Computing*, vol. 7, no. 1, pp. 74–81, 2008.

[14] N. Bandirmali and I. Ertürk, "Increasing the reliability of security protocols for WSNs," in *Proceedings of the International Conference on Application of Information and Communication Technologies (AICT '09)*, Baku, Azerbaijan, October 2009.

[15] S. Ozdemir, "Wireless sensor network security: a comprehensive overview," *Journal of Politecnic*, pp. 217–244, 2008.

[16] E. Shi and A. Perrig, "Designing secure sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 38–43, 2004.

[17] J. Sen, "Security in wireless sensor networks," in *Wireless Sensor Networks: Current Status and Future Trends*, 2012.

[18] K. Venkatraman, J. Vijay Daniel, and G. Murugaboopathi, "Various attacks in wireless sensor network: survey," *International Journal of Soft Computing and Engineering*, vol. 3, no. 1, 2013.

[19] C. Karlof and D. Wagner, "Secure routing in wireless SensorNetworks: attacks and countermeasures," in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 113–127, May 2003.

[20] V. Soni, P. Modi, and V. Chaudhri, "Detecting Sinkhole attack in wireless sensor network," *International Journal of Application or Innovation in Engineering & Management*, vol. 2, no. 2, 2013.

[21] G. Padmavathi and D. Shanmugapriya, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," *International Journal of Computer Science and Information Security*, vol. 4, no. 1-2, 2009.

[22] T. K. Rao, M. Sharma, and M. V. Saradhi, "Wormhole attacks in Ad-Hoc networks," *International Journal of Latest Trend in Computing*, vol. 4, no. 2, 2013.

[23] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 2, pp. 2–22, 2006.

[24] H. C. Chaudhari and L. U. Kadam, "Wireless sensor networks: security, attacks and challenges," *International Journal of Networking*, vol. 1, no. 1, pp. 4–16, 2011.

[25] A. Poschmann, G. Leander, K. Schramm, and C. Paar, "New light-weight DES variants suited for RFID applications," in *Proceedings of 14th Annual Fast Software Encryption Workshop (FSE '07)*, Luxembourg, Germany, March 2007.

[26] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (Blowfish)," in *Fast Software Encryption*, vol. 809 of *Lecture Notes in Computer Science*, pp. 191–204, Springer, Berlin, Germany, 1994.

[27] D. J. Wheeler and R. M. Needham, "TEA (tiny encryption algorithm)," in *Proceedings of Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994*, vol. 1008 of *Lecture Notes in Computer Science*, pp. 363–366, 1994.

[28] T. Li, H. Wu, X. Wang, and F. Bao, *SenSec Design Technical Report-TR v1.1*, InfoComm Security Department, Institute for Infocomm Research, 2005.

[29] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 162–175, Baltimore, Md, USA, November 2004.

[30] 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, ACM, April 2007.

[31] F. X. Standaert, G. Piret, N. Gershenfeld, and J. J. Quisquater, "SEA: a scalable encryption algorithm for small embedded applications," in *Workshop on RFIP and Light weight Crypto*, Graz, Austria, 2005.

[32] D. Hong, "HIGHT: a new block cipher suitable for low-resource device," in *Cryptographic Hardware and Embedded Systems—CHES 2006: 8th International Workshop, Yokohama, Japan, October 10–13*, vol. 4249 of *Lecture Notes in Computer Science*, pp. 46–59, Springer, Berlin, Germany, 2006.

[33] W. Stallings, *Cryptography and Network Security*, Prentice Hall, 4th edition, 2005.

[34] H. Kodaz, *Cryptography for security on data communication [M.S. thesis]*, Selçuk University, Institue of Science and Technology, 2002.

[35] 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.

[36] T. Park and K. G. Shin, "LiSP: a lightweight security protocol for wireless sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 3, pp. 634–660, 2004.

[37] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[38] IEEE-TG15.4, *PART 15.4:Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Standard for Information Technology, 2003.

[39] A. Koubaa, M. Alves, and E. Tovar, "IEEE 802.15.4: a federating communication protocol for time-sensitive wireless sensor networks," in *Sensor Networks and Configurations: Fundamentals, Tecniques, Platforms and Experiments*, pp. 19–49, Springer, Berlin, Germany, 2007.

[40] R. A. Shaikh, S. Lee, M. A. Khan, and Y. C. Song, "LSec: lightweight security protocol for distributed wireless sensor network," in *Personal Wireless Communications*, vol. 4217 of *Lecture Notes in Computer Science*, pp. 367–377, 2006.

[41] S. Tripathy, "LISA: lightweight security algorithm for wireless sensor networks," in *Proceeding of Distributed Computing and Internet Technology, 4th International Conference, ICDCIT 2007, Bangalore, India, December 17-20*, vol. 4882 of *Lecture Notes in Computer Science*, pp. 129–134, 2007.

[42] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: a secure sensor network communication architecture," in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN '07)*, pp. 479–488, April 2007.

[43] ZigBee Alliance, "Zigbee specification," Technical Report Document 053474r06, Version 1.0, ZigBee Alliance, 2005.

[44] L. E. Lighfoot, J. Ren, and T. Li, "An energy efficient link-layer security protocol for wireless sensor networks," in *Proceedings of the IEEE International Conference on Electro/Information Technology (EIT '07)*, pp. 233–238, Chicago, Ill, USA, May 2007.

[45] H. Kodaz, *Cryptography for security on data communication [M.S. thesis]*, Selçuk University Institue of Science and Technology, 2002.

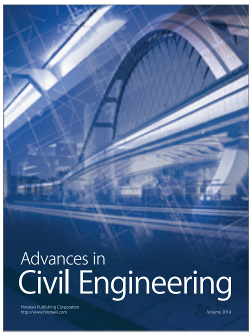
Journal of
Engineering

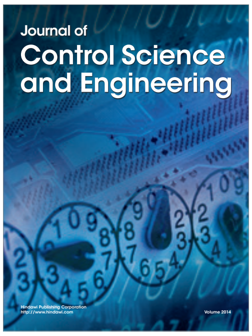
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

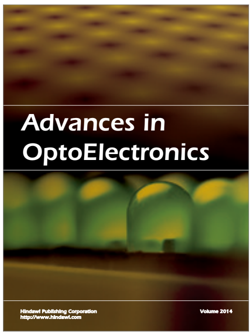
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

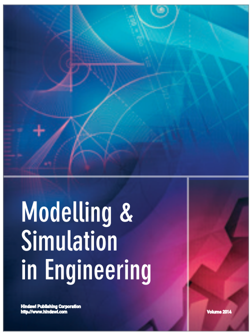
Journal of
Robotics

Journal of
Electrical and Computer Engineering

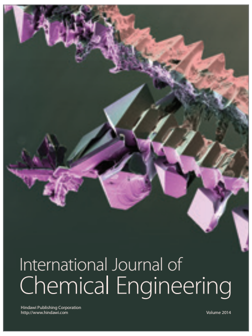

Advances in
OptoElectronics

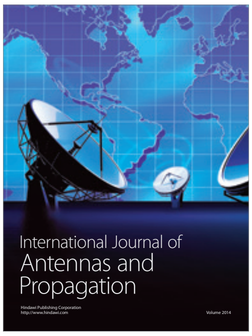
VLSI Design

International Journal of
Navigation and Observation

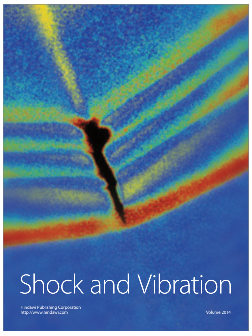

Modelling & Simulation in Engineering

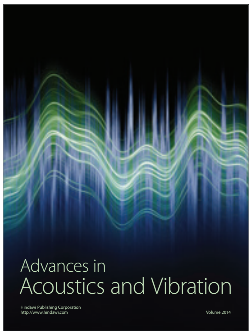
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration