International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

A Security Key Management Method using Grid Routing for EDDK in WSNs

Won Jin Chung and Tae Ho Cho

Abstract*: Wireless sensor networks (WSNs) utilize event data collected through sensor nodes, which can be employed in different environments. Sensor nodes used in WSNs have limited energy and can be easily compromised by attackers since they are placed in the external environment. Therefore, energy efficiency and security are core requirements in WSNs, and various schemes have been proposed to solve them. The energy-efficient distributed deterministic key management scheme (EDDK) is a security scheme that uses distributed pairwise keys. The EDDK encrypts the packet using the pairwise key of the neighboring node, and transmits the encrypted packet securely to the base station (BS). In addition, the pairwise key shared with neighboring nodes is updated to defend against network attacks such as Denial of Service. However, the energies of the sensor nodes decrease rapidly in environments where key updates occur frequently. Further, since EDDK does not encrypt the sequence number, the attacker can manipulate this sequence number. If such attacks are continually attempted, key updates occur frequently. This paper proposes a scheme that removes the local cluster key from the EDDK and instead uses a grid routing protocol. The proposed scheme reduces the number of key updates by removing the local cluster key. In addition, the proposed scheme constructs the routing as a grid and selects the sensor node nearest to the BS as the master node. the master node of the grid routing scheme transmits routing control messages to the corresponding sensor nodes. In this way, the proposed method improves the sensor network energy efficiency. Experimental results show that the energy efficiency of the sensor network is improved by about 7.8097% compared with EDDK. Additionally, since the proposed scheme removes the local cluster key, sequence number manipulation attacks can be avoided. As a result, the security rate of the proposed scheme is more than double compared with EDDK.

Keywords* : Grid Routing, Key management, Network Security, Wireless Sensor Networks

I. INTRODUCTION

Wireless sensor networks (WSNs) are used for real-time monitoring of ecosystem status or enemy intrusion detection through sensors. WSNs are used in various fields requiring sensing data, such as industrial, mining, and disaster management. WSNs consist of sensor nodes that collect information or detect desired events such as sound, temperature, and vibration, after data collection, a base station (BS) then analyses this collected information. The sensor nodes monitor the external environment to detect events, and when an event is detected, the nodes generate an event packet and transmit it to the BS via wireless communication between the sensor nodes. The BS collects the information received from the sensor nodes and transmits this information to the user through the internet and the satellite. Fig. 1 shows the structure of a WSN [1].

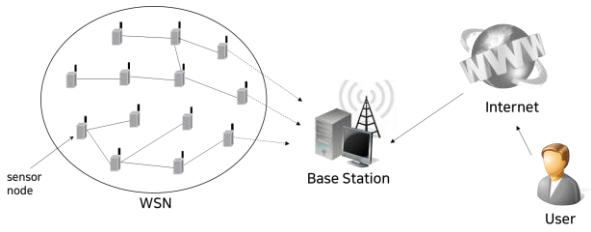

Fig. 1. Structure of a WSN

Since WSNs use low-cost sensor nodes, many sensors can be deployed in a large area to accurately detect events. Since sensor nodes transmit event information to the BS through wireless communication, a user can remotely collect the desired information for various purposes. However, sensor nodes deployed in a WSN have limited performance in terms of their battery life, communication distance, memory, and computing power. The sensor nodes are operated with a battery, and sensor nodes are difficult to recharge once deployed. Thus, battery limitations of the sensor nodes are among the biggest problems in WSNs. If many sensor nodes are depleted of energy, it is impossible to transmit the detected event information to the BS. Sensor nodes use low-power consumption communication techniques such as ZigBee communication because of their limited energy. However, since ZigBee communication has a short transmission distance, in order to monitor a wide area, many sensor nodes participate in communication and transmission to the BS. Since sensor nodes are small in size, small-sized memory cards are used. Therefore, not much data can be stored in the memory of a sensor node, and so only essential information such as routing information is stored and used. In addition, sensor nodes used in WSNs are vulnerable to external attacks since they are placed in an external environment and transmit data to the BS through wireless communication. Sensor nodes that are placed outside are easily compromised by malicious attackers because of their limited computing power and memory performance [2][3].

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author
Won Jin Chung*, Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Republic of Korea. Email: wonjin12@skku.edu
Tae Ho Cho, Department of Software Platform, Sungkyunkwan University, Suwon, Republic of Korea. Email: thcho@skku.edu

The attacker uses a corrupted node to generate a packet containing false content, encrypts it as a normal packet using the key information, and then transmits it to the other sensor nodes. Further, since the attacker knows the key information, they can analyze the encryption packet transmitted through the compromised node and attempt to forge or delete the packet contents. Therefore, it is very important to protect the key information of the sensor nodes, and various security scheme have been studied. The energy-efficient distributed deterministic key management scheme (EDDK) proposed by Xing Zhang et al. focuses on the establishment and maintenance of pairwise keys and local cluster keys [4]. EDDK has many advantages over other similar schemes such as the localized encryption and authentication protocol (LEAP) [5][6] and opaque transitory master key (OTMK) schemes [7], particularly in terms of computation, communication, energy efficiency, and storage space management. However, key updates frequently occur in environments where events occur frequently, where the energies of the sensor nodes are exhausted, or where sensor nodes are vulnerable to attacks. If key updates occur frequently, the neighboring node is continually consuming energy for communication. If a sensor node with a small amount of residual energy is selected as a neighbor node when key updates are frequently generated, energy consumption is increased in the neighbor node. If this is repeated enough times, energy exhaustion will occur, thus shortening the life of the WSN. Therefore, the proposed scheme removes the local cluster key used in EDDK to reduce energy consumption for key updates, and improves the energy efficiency by efficiently transmitting keys through grid routing. The composition of this paper is as follows. Section 2 describes common types of attacks on the network layers of WSNs, on EDDK for the network layer security protocol, and on the grid-based multipath with congestion avoidance routing protocol (GMCAR) for routing. Section 3 describes the motivation and assumptions of the proposed scheme and describes how the proposed scheme functions in detail. Section 4 describes the experimental environment and the results. Finally, Section 5 discusses conclusions and future research.

II. RELATED WORKS

A. *Networks Layer Attacks*

Since the sensor nodes used in WSNs are vulnerable to physical security, an attacker can attack these nodes using such techniques as node replication attacks, destroying the sensor nodes, and distributing damaged sensor nodes to the network. Then, using the compromised nodes, the normal packet can be dropped to prevent it from reaching the BS, or the contents of the packet can be modified before transmission to the BS. Denial of Service (DoS), Distributed DoS (DDoS), Sybil, Sinkhole, Blackhole and Selective forwarding attacks are typical approaches to attacking the network layer [2]. A DoS or DDoS attack generates false packets from the compromised node and delivers them to the sensor nodes included in the path. A sensor node that receives too many packets can drop packets that are critical to processing packets transmitted from a compromised node, even when a packet containing important information is transmitted. In this way, such critical packets are not transmitted to the BS, and the energy of the sensor node is continually consumed. When a sensor node sets a path, the compromised node informs other sensor nodes of position information not included in the path. When an event occurs, an event notification packet is generated and transmitted. In the case of a sybil attack, a packet is transmitted through an incorrect path instead of a normal path, and important packets are dropped. A node experiencing a sinkhole attack communicates to the other sensor nodes that it has the shortest path. An attacker may then attempt additional attacks, such as selective forwarding and blackhole attacks, using this compromised node. A compromised node may cause confusion by deleting or selectively transmitting packets. To defend against such network attacks, various security schemes have been studied. In addition, since energy is consumed when a packet is applied in a security scheme, studies to increase the energy efficiency of a network have been conducted.

B. *Energy-efficient Distributed Deterministic Key management scheme (EDDK)*

EDDK protects the key of the sensor nodes from an attacker using a Pairwise key, Local Cluster key, Public key, and Private key. EDDK consists of a key establishment phase, data transfer phase, and key maintenance phase. Fig. 2 shows the overall process of EDDK.

Fig. 2. EDDK process

The key establishment phase consists of deploying sensor nodes in the sensor field and establishing the pairwise key with the neighbor nodes. The sensor nodes store the pseudo-random function f [8] and initial key K_I, which are distributed throughout the sensor field. Then, the sensor nodes compute the individual keys of the sensor nodes using f and K_I. Since an individual key is calculated using the sensor node ID, the individual keys of the sensor nodes placed in the WSNs have different values. Calculation of the individual key for sensor node a is shown in Equation 1.

$$K_a = f(K_I, ID_a) \quad (1)$$

Here, K_a is the individual key of sensor node a, and ID_a is the ID of sensor node a. EDDK derives the pairwise key and message authentication code (MAC) key using the individual keys of the nodes to improve security [9]. The sensor node searches the neighbor nodes and establishes pairwise keys between the neighbor nodes. The pairwise key setting of sensor node a and sensor node b is shown in Equation 2.

$$K_{ab} = f(K_a \oplus K_b, SN_a \oplus SN_b) \quad (2)$$

Here, K_{ab} is the pairwise key between sensor node a and sensor node b, and SN_a and SN_b are random numbers generated by each sensor node to distribute the pairwise key. After the pairwise key setup is completed, the pseudo-random function and initial key are deleted for security enhancement. Since the sensor nodes deployed in the sensor field delete the stored pseudo-random function and initial key, the attacker cannot compute the key information (i.e., the individual keys and pairwise keys) of a sensor node even if the attacker has successfully compromised the sensor node. A local cluster key is a key that a sensor node shares with all its neighbors, and is periodically updated to improve security. This key is used to protect the local broadcast messages of the sensor nodes, such as routing control messages. We now consider the data transfer phase. The neighbor table of sensor nodes is shown in Table 1.

Table 1. Neighbor table

2 Bytes	8 Bytes	8 Bytes	2 Bytes	2 Bytes
Node ID	Pairwise Key	Local Cluster Key	Pairwise Key Sequence Number	Local Cluster Key Sequence Number

The pairwise key and the local cluster key each have their own sequence, and this sequence serves as a key lifetime. A sequence is initially set to 0, and upon reaching a pre-selected threshold, performs a key update, after which the sequence is initialized back to 0. In addition, such a sequence can protect packets from retransmission attacks because the number is different each time a packet is delivered. EDDK does not verify every packet in order to improve the energy efficiency. When a sensor node receives a packet, it checks the ID and sequence number of the corresponding sensor node in the neighbor table. The MAC is calculated only when the neighbor table and the packet information are the same. If the packet information and the neighbor table information are different, EDDK drops instead of calculates the MAC, thus reducing the energy consumed during verification. If the MAC of the received packet matches the MAC generated by the sensor node, the packet received is considered a normal packet. After that, the sensor node generates a new MAC and transmits it to the next sensor node. The key maintenance phase provides overall management of the network regarding key updates, compromised key revocation, and new node joining. Key updates utilize the sequence of each key to determine the update period. Compromised key revocation isolates a compromised node from the network and disables any pairwise keys shared with this compromised node. After completing this step, the normal sensor nodes update the local cluster key. New node joining utilizes the public key and timestamp. This timestamp is used to identify the new nodes. The sensor node checks the timestamp when adding a new node to the path. If the time of addition is different from that of the timestamp, the added node is suspected to be a compromised node. In addition, the sensor nodes can verify whether the new node is a normal node by using the Elliptic Curve Digital Signature Algorithm (ECDSA) [10-12]. ECDSA is a public key-based verification method similar to the Rivest Shamir Adleman (RSA) scheme [13][14], which is a representative public key cryptosystem. However, ECDSA consumes less energy compared to RSA because it uses an elliptic curve. Thus, ECDSA is an appropriate scheme available for use in WSNs.

C. *Grid-based Multipath with Congestion Avoidance Routing (GMCAR) protocol*

Routing protocols are among the most important communication paradigms in WSNs [15]. Therefore, it is important for the user to design efficient routing considering the resource constraints (e.g., energy, communication range, and computing power) of the sensor node and communication delay [16-18]. The Grid-based Multipath with Congestion Avoidance Routing (GMCAR) protocol is a grid-based multipath routing protocol that supports quality of service (QoS) traffic for WSNs [19]. WSNs have constraints regarding the support of multiple classes of traffic, delay energy trade-offs, reliability versus redundancy, multipath routing, and network congestion while providing good QoS. To mitigate these issues, GMCAR uses grid routing to efficiently transmit data. GMCAR is classified as multipath because it establishes multiple diagonal paths between all master nodes and the BS. GMCAR consists of the grid formation phase, routing table construction phase, and data transmission phase. The grid formation phase divides the sensor field into logical square-shaped grids with a predefined size. Because sensor nodes are typically randomly placed, situations can arise where only a single node is located in the partitioned grid. In this case, this single node is selected as the master node. When one or more sensor nodes are placed in the grid, the master node is determined as the sensor node with the highest ID among the sensor nodes arranged in the grid. In the routing table construction phase, the master node selected in the previous step transmits a flooding message for routing to the BS. The grid is divided into boundary and a non-boundary sections according to the position of the BS, and the position of the BS is located at one of following grid locations: upper left, upper right, lower left, or lower right. Fig. 3 shows the boundary and non-boundary sections of the GMCAR protocol. The grid portions around the BS belong to the boundary and the packet is transmitted in a straight line. In addition, the grid portions included in the non-boundary comprise a protocol that efficiently transmits packets diagonally and to the grid portions of the boundary.

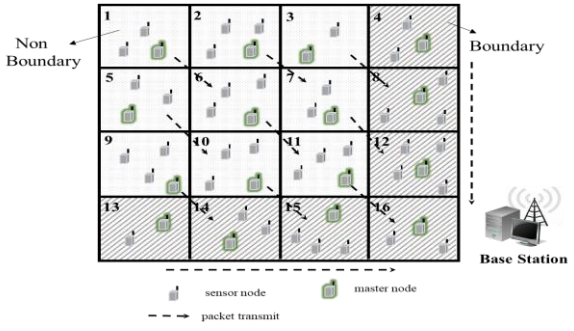

Fig. 3. GMCAR protocol

The flooding message is broadcasted from the BS and then re-broadcasted after updating the grid information at each sensor node. By repeating this process, each sensor node creates a routing table. Finally, after setting up the routing table, the master node can begin data transmission to the BS. When an event occurs, the non-master node that detected the event transmits a packet to the master node of the same grid, and the master node that received the packet selects the master node of the next grid to forward the packet. Most of the sensor nodes not involved in communication are switched to sleep mode, while the master node remains active and waits for packet transmissions. Switching the sensor nodes to the sleep mode improves the energy efficiency of the WSN. When the residual energy of the master node is almost exhausted, a new master node is selected. Specifically, the sensor node with the highest residual energy among the non-master nodes is selected as the new master node.

III. PROPOSED SCHEME

A. *Motive*

EDDK periodically performs pairwise key and local cluster key updates. The key (e.g., pairwise keys and local cluster keys) update period is shorter in areas where there are many compromised nodes and in environments where many events occur. Such an environment continually consumes the energy of the sensor nodes, and thus energy depletion of sensor nodes must be considered. Additionally, because EDDK uses a pairwise key and a local cluster key, an attacker can attempt to manipulate the sequence. EDDK does not encrypt these sequences to reduce energy consumption for packet verification. Thus, an attacker can reduce the energy of a sensor node through a replay attack on a sequence using this vulnerability.

B. *Assumption*

The sensor nodes are arranged randomly and there is at least one sensor node in the grid. The master node is selected as the sensor node that is closest to the BS. Further, if the energy of the current master node is depleted, the next non-depleted sensor node closest to the BS is selected as the new master node. We assume that the BS is not attacked and can locate each sensor node. The residual energies and states of the sensor nodes are known by the BS.

C. *Overview*

The proposed scheme reduces the energy consumption for key updating by removing the local cluster key from EDDK. Additionally, by implementing GMCAR, master nodes are selected and efficient transmission of event packets to the BS is achieved. In EDDK, the routing control messages encrypted with the local cluster key must be transmitted to the neighbor nodes in a different way; this is because the proposed scheme does not consider a local cluster key. Instead, in the proposed scheme, a sensor node that needs to change routing transmits a routing control message to the corresponding non-master node after encrypting this message via its pairwise key with the master node. This method leads to faster energy consumption of the master node relative to the other nodes, but keeps the transmission more secure. In EDDK, the routing control message is encrypted via a local cluster key and broadcasts it to the neighbor nodes. Even if the neighbor nodes do not need to change the local cluster key, they process the packet and thus consume energy. However, in the proposed scheme, network energy efficiency is improved because the routing control message is transmitted as a unicast to the neighbor nodes using the pairwise key of the master node. Since EDDK transmits the sequence used in the packet key without encrypting it, this sequence is vulnerable to attack. An attacker can shorten the key update cycle by manipulating the sequence number, thus more energy is consumed than normal because the update cycle becomes faster. In an environment with high traffic, such an attack can significantly affect the energy depletion factor of the sensor nodes. Although the proposed scheme does not completely protect from sequence number attacks, it decreases the probability of attack by removing the local cluster key. Therefore, compared with EDDK, the security rate of the proposed method is improved.

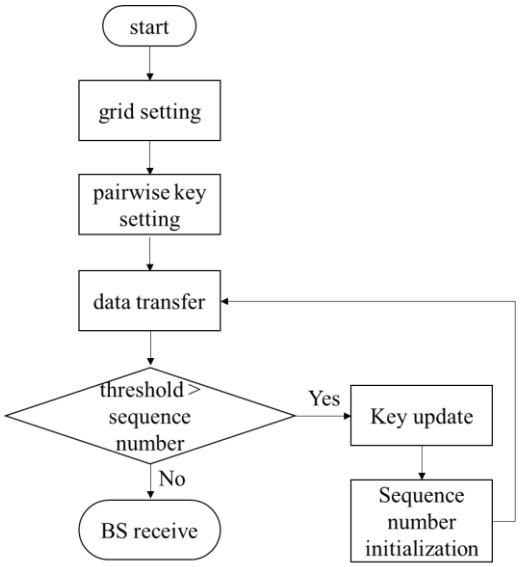

Fig. 4. Key update flowchart of the proposed scheme

Fig. 4 shows the key update process for the proposed scheme. The proposed scheme transmits data to the BS without setting up a local cluster key. However, similar to EDDK, the key update process proceeds if the sequence number is greater than a preset threshold, after which the sequence number is initialized.

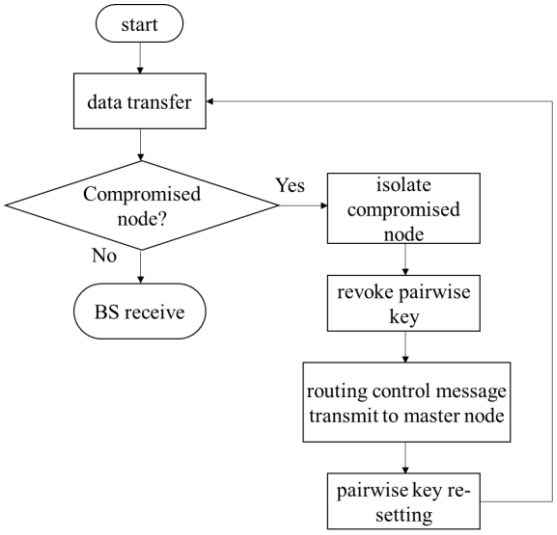

Fig. 5. Compromised node detection flowchart of the proposed scheme

Fig. 5 shows the procedure taken in the proposed scheme when a compromised node is found in the sensor field. The proposed scheme isolates the compromised node similar to EDDK, and drops the pairwise key connection. In the EDDK scheme, the routing control message is sent to the sensor nodes using the local cluster key. However, in the proposed scheme, the routing control message is sent instead to the master node of the grid routing using pairwise keys. When the master node receives the routing control message, it resets the pairwise keys of the sensor nodes connected to the compromised node and forwards the data to the BS.

IV. EXPERIMENTAL RESULTS

In this section, we simulate the packet transmission and attack processes for routing after sensor node deployment in a WSN using the proposed scheme. The simulations were written in C++ using Visual Studio. Table 2 shows the parameters and values used in the simulation environment.

Table 2. Parameters used in the simulation

Simulation parameter	Value
Sensor field size	200 m × 200 m
Number of grids	16
Sensor node	MICAz
Number of sensor node	100
Initial sensor node energy	Random
Maximum sensor node energy	1 J
Radio range	150 m
Data packet size	128 bytes

The sensor node is designed according to the MICAz specifications. The energies of the sensor nodes are set randomly to consider more general environments, but do not exceed 1J. MICAz consumes 0.60μJ per bit when transmitting packets and 0.72μJ per bit when receiving packets. MICAz consumes 9.2nJ of energy per clock cycle in listen mode and 3pJ in sleep mode [20]. The CBC-MAC is generated using RC5 (with 12 rounds) under the same conditions as EDDK. At the sensor nodes, the energy consumed to generate RC5 is 49.92μJ per cycle [21]. The number of events used in the experiment does not exceed 1000.

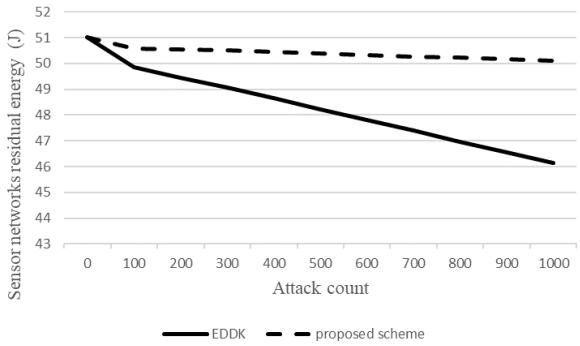

Fig. 6. Sensor network efficiency (attack count)

Fig. 6 shows the residual energy of the sensor nodes with respect to the frequency of events with an attack rate of 60%. For 1000 events, the proposed scheme improves the energy efficiency by about 7.8097% compared with EDDK.

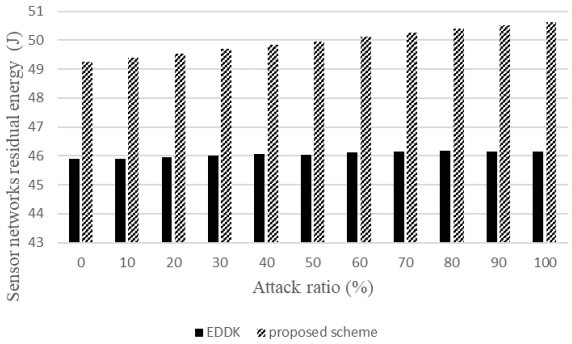

Fig. 7. Sensor network efficiency (attack ratio)

Fig. 7 shows the residual energy of the sensor nodes with respect to the attack rate in an environment for 1000 events. A similar trend in energy efficiency was exhibited between EDDK and the proposed scheme. However, since the proposed scheme does not consider a local cluster key, there is no energy consumption for updating the local cluster key, resulting in overall lower energy consumption. In addition, since the routing control message is transmitted using pairwise keys and grid routing, it consumes less energy than EDDK. Therefore, the proposed scheme has improved energy efficiency compared with EDDK.

Table 3. Sequence number initialization count

Attack ratio (%)	Sequence number initialization count	
	EDDK	Proposed scheme
0	549	256
10	556	257
20	569	264
30	575	266
40	598	272
50	613	280

60	622	284
70	630	258
80	630	268
90	658	297
100	691	309

Table 3 shows the number of sequence number initializations with respect to the attack rate for 1000 events, comparing the proposed scheme with EDDK. Because EDDK does not encrypt the sequence number, the attacker can manipulate the sequence number and shorten the key update period. Manipulation of the sequence number cannot be completely defended against when using the proposed scheme. However, since the proposed scheme does not consider the local cluster key, it can avoid sequence manipulations related to this local cluster key. Thus, the proposed scheme exhibits an improved security rate compared to EDDK, about twice as much as EDDK, as seen in Table 3.

V. CONCLUSIONS

EDDK is a security scheme that protects against attacks in the network layer, which is accomplished by focusing on the establishment and maintenance of pairwise keys and local cluster keys. However, in an environment where a large number of compromised nodes are deployed or where a large number of events occur, key updates frequently occur. The energies of the sensor nodes then rapidly decrease due to these frequent key updates until the energies of the sensor nodes are exhausted. If there are too many energy exhausted sensor nodes, the network life is shortened. The proposed scheme improves the energy efficiency and security of the sensor networks by using grid routing and removing the local cluster key. Removing the local cluster key reduces the number of key updates and avoids sequence manipulation attacks. Furthermore, the proposed scheme encrypts the routing control message using the pairwise key of the master node, and transmits the message to the corresponding sensor node. This method can transmit messages using grid routing without a local cluster key. Experimental results show that the energy efficiency of the sensor network is improved by about 7.8097% when 1000 events are generated compared with EDDK, and the security rate for sequence operations is about twice as high as that of EDDK. However, the proposed scheme does not provide a complete defense against sequence number manipulation attacks. Future research will focus on EDDK's security against sequence number manipulation attacks, as well as enhancing security by considering multiple simultaneous attacks occurring in the network layer.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. NRF-2018R1D1A1B07048961)

REFERENCES

1. I. F. Akyildiz, et al. "Wireless sensor networks: a survey." Computer networks vol. 38, no. 4, pp. 393-422, 2002.
2. C. Karlof and D. Wagner. "Secure routing in wireless sensor networks: Attacks and countermeasures." Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003. IEEE, 2003.
3. Y. Wang, G. Attebury, and B. Ramamurthy. "A survey of security issues in wireless sensor networks." IEEE Communications Surveys & Tutorials 2nd Quarter 2006.
4. X. Zhang, J. He, and Q. Wei. "EDDK: Energy-efficient distributed deterministic key management for wireless sensor networks." EURASIP Journal on Wireless Communications and Networking 2011, no. 12, 2011.
5. S. Zhu, S. Setia, and S. Jajodia, “LEAP: efficient security mechanisms for large-scale distributed sensor networks,” in Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS ’03), pp. 62–72, Washington, DC, USA, October 2003.
6. S. Zhu, S. Setia, and S. Jajodia, “LEAP+: efficient security mechanisms for large-scale distributed sensor networks,” ACM Transactions on Sensor Networks, vol. 2, no. 4, pp. 500–528, 2006.
7. J. Deng, C. Hartung, R. Han, and S.Mishra, “A practical study of transitory master key establishment for wireless sensor networks,” in Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm ’05), pp. 289–299, Athens, Greece, September 2005.
8. O. Goldreich, S. Goldwasser, and S. Micali, “How to construct random functions,” Journal of the ACM, vol. 33, no. 4, pp. 792–807, 1986.
9. SM Chang, et al. "An efficient broadcast authentication scheme in wireless sensor networks." Proceedings of the 2006 ACM Symposium on Information, computer and communications security. ACM, 2006.
10. D. Johnson, A. Menezes, and S. Vanstone. "The elliptic curve digital signature algorithm (ECDSA)." International journal of information security, vol. 1, No. 1, pp. 36-63, 2001.
11. A. S.Wandert, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, “Energy analysis of public-key cryptography for wireless sensor networks,” in Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom ’05), pp. 324– 328, Kauai Island, Hawaii, USA,March 2005.
12. G. Gaubatz, J.-P. Kaps, and B. Sunar, “Public key cryptography in sensor networks-revisited,” in Proceedings of the 1st EuropeanWorkshop on Security in Ad-Hoc and Sensor Networks, pp. 2–18, Heidelberg, Germany, August 2005.
13. R. L. Rivest, A. Shamir, and L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems." Communications of the ACM, Vol. 21, No. 2 pp. 120-126, 1978.
14. N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, “Comparing elliptic curve cryptography and RSA on 8-Bit CPUs,” in Proceedings of the 6th International workshop on Cryptographic Hardware and Embedded Systems (CHES ’04), pp. 119–132, Cambridge, Mass, USA, August 2004.
15. JN. Al-Karaki and AE. Kamal. "Routing techniques in wireless sensor networks: a survey." IEEE wireless communications vol. 11, no. 6, pp. 6-28, 2004.
16. M. Younis, K. Akkaya, and M. Youssef. "Handling qos traffic in wireless sensor networks." Encyclopedia on Ad Hoc and Ubiquitous Computing: Theory and Design of Wireless Ad Hoc, Sensor, and Mesh Networks, pp. 257-279, 2010.
17. K. Akkaya and M. Younis. "Energy and QoS aware routing in wireless sensor networks." Cluster computing vol. 8, no. 2-3, pp. 179-188, 2005.
18. R. Akl and U. Sawant. "Grid-based coordinated routing in wireless sensor networks." 2007 4th IEEE Consumer Communications and Networking Conference. IEEE, 2007.
19. O.Banimelhem and S.Khasawneh. "GMCAR: Grid-based multipath with congestion avoidance routing protocol in wireless sensor networks." Ad Hoc Networks, vol. 10, no. 7, pp. 1346-1361, 2012.
20. G. D. Meulenaer, et al. "On the energy cost of communication and cryptography in wireless sensor networks." 2008 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications. IEEE, 2008.
21. G. Germano, et al. "Evaluation of security mechanisms in wireless sensor networks." 2005 Systems Communications (ICW'05, ICHSN'05, ICMCS'05, SENET'05). IEEE, 2005.

AUTHORS PROFILE

Won Jin Chung Received a B.S. degree in Information Security from Baekseok University, Korea, in 2016 and is now working toward a Ph.D. degree in the Department of Electrical and Computer Engineering at Sungkyunkwan University, Korea.

Tae Ho Cho Received a Ph.D. degree in Electrical and Computer Engineering from the University of Arizona, USA, in 1993, and B.S. and M.S. degrees in Electrical and Computer Engineering from Sungkyunkwan University, Republic of Korea, and the University of Alabama, USA, respectively. He is currently a Professor in the College of Software, Sungkyunkwan University, Korea.