Detection of Replay Attack through Sequence Number Encryption in EDDK based WSNs

Won Jin Chung, Tae Ho Cho

Abstract**: *Wireless sensor networks can be used to deliver status information to users in real time. The sensor nodes used in wireless sensor networks are arranged by attaching sensors to acquire necessary information, such as vibration, sound, light, and temperature. Since a sensor node is small in size and inexpensive, it is advantageous for monitoring large areas. When a sensor node senses a change in a situation, this event information is wirelessly communicated with other sensor nodes and transmitted to a base station. However, since the sensor nodes used in wireless sensor networks are small and inexpensive, there are restrictions in terms of their battery life, memory, and computing power. An attacker can easily compromise a sensor node and use a compromised node to attempt message tampering and energy consumption attacks. EDDK is a scheme that detects attacks from compromised nodes through key management. EDDK uses a pairwise key and a local cluster key to defend against various attacks in the network. In addition, EDDK protects against replay attacks by using sequence numbers and guarantees message integrity. However, since the sequence number and sensor node ID are not encrypted, they can be used as an attack element. An attacker can attempt a replay attack by manipulating the sequence number through sniffing. A replay attack that occurs in a wireless sensor network consumes sensor node energy and confuses the user. In order to defend against such an attack, we propose a sequence number encryption scheme. The proposed scheme detects new types of replay attacks and shows about 7% energy improvement.***

Keywords**: *network security, replay attack, sequence number management, wireless sensor networks.***

I. INTRODUCTION

Modern society can easily obtain information from desired locations due to the development of electronic information and communication instruments. This necessary information can be acquired through a sensor module mounted on a device. This sensor module can collect necessary information, such as information related to light, gas, sound, pressure, and temperature. As the sensor's hardware performance improves, more accurate detection is allowed.

Revised Manuscript Received on July 30, 2020.
***** 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 Computer Science and Engineering, Sungkyunkwan University, Suwon, Republic of Korea. Email: thcho@skku.edu

Wireless sensor networks (WSNs) are a technology that can deliver necessary information to users using these sensors [1]. A WSN consists of small sensor nodes and base stations (BSs). The number of BSs needed depends on the environment in which the sensor nodes are deployed. A sensor node is equipped with a sensor module (for detecting an external situation) and a processor and memory (for data processing). In addition, a sensor node has a wireless transceiver (for transmitting and receiving data) and a battery (to supply energy). Since sensor nodes are small and inexpensive, they are arranged in a large area to collect information about the surrounding environment and provide object recognition information in real time. A sensor node transmits this collected information to a BS through wireless communication. The BS stores, manages, and analyzes information collected through sensor nodes. Subsequently, the BS transmits the processed information to the user. Users can use this information in a variety of fields. Figure 1 shows the process of delivering an event detected by a sensor node to a user.

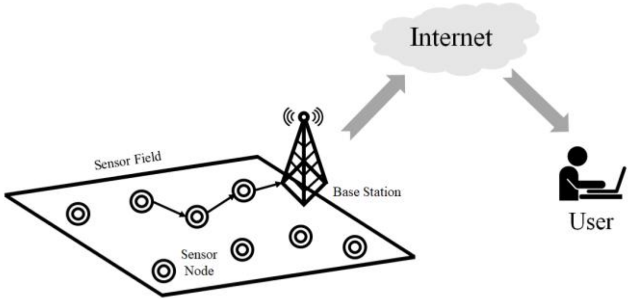

Fig. 1. Event Delivery Process Through a Sensor Node

In the past, WSNs were developed to monitor battlefields. Currently, these networks aim to improve user convenience by providing monitoring information along with the Internet of Things. WSNs provide continuous monitoring information and are placed in hazardous areas that are difficult to access. They can also be deployed along the coast to detect tsunamis, providing continuous monitoring that can be used to predict natural disasters. For example, a system has been proposed to measure the intensity of earthquakes by installing seven sensor nodes and a sink node [2]. In addition, WSNs are deployed in bridges, tunnels, child protection areas, industrial parks, etc., and are used in various fields to monitor human accidents and safety. Although WSNs are used in various fields, there are restrictions on the computing power, main memory, communication protocol, and battery capacity of sensor nodes. First, the size of the main memory is limited because sensor nodes are small. Therefore, only essential information, such as network and routing information, is stored in the memory of a sensor node. Since sensor nodes have low memory, it is necessary to meet these constraints when designing the accompanying security protocol. The next issue is related to the battery problems faced by sensor nodes.

When a sensor node is placed in the field, it acts as a monitoring device until the end of its battery life. When the energy of a sensor node is depleted, other sensor nodes in the vicinity perform this monitoring role instead. Due to this problem, many sensor nodes must be deployed. Also, if the energy of all sensor nodes deployed in a certain area is exhausted, event detection is impossible in that area. With the development of embedded systems and wireless communication technology, various technologies related to microminiaturization have been studied. However, a sensor node needs to be configured at a low cost; thus, they have limited performance. Sensor nodes used in WSNs are vulnerable to security issues due to these performance limitations. A sensor node is placed outside and transmits information using wireless communication. An attacker can compromise a sensor node with an attack, such as a clone attack. After that, the attacker attempts various attacks that can damage the WSN using the compromised node. An attacker can use the compromised node to monitor packets to obtain event information or to falsify data, which can confuse the user. Another type of attack is to transmit false information from a compromised node [3], [4]. Such an attack consumes the sensor node’s energy through unnecessary message transmission, thereby depleting the energy of the sensor node as the attack continues. WSNs with limited energy must defend against energy consumption attacks. In order to defend against such attacks, many security schemes have been proposed. Among the various security techniques, the energy-efficient distributed deterministic key management scheme (EDDK) proposed by Xing Zhang focuses on the pairwise key and local cluster key settings [5]. EDDK is a routing-related security technique using security keys, such as initial key establishment, new node joining, and withdrawing a compromised node. EDDK showed higher energy efficiency than the localized encryption and authentication protocol (LEAP) and opaque transitory master key (OTMK) schemes [6], [7]. In addition, EDDK has an advantage of securing a storage space by using a method to generate a pairwise key and remove the used key. EDDK does not use a master node, so it is more secure than other schemes. EDDK performs periodic updates to prevent the previous message from being captured, even if the key is stolen by an attacker. The update cycle is designed to run when the sequence number reaches a predetermined threshold. EDDK can use the sequence number to determine the key's lifetime and save storage space. The sequence number not only determines the lifetime of the pairwise key, but it also ensures the freshness of the data to prevent replay attacks. Also, it is possible to drop a false message early by checking only the sequence number and the sensor node ID. However, the attacker can confirm that the sequence number is not encrypted by sniffing the data; when this happens, the attacker can attempt a retransmission attack by changing the sequence number. Since EDDK defends against replay attacks using sequence numbers, attack detection cannot be performed if sequence numbers are manipulated. In addition, since the rest of the messages (except the sequence number and sensor node ID) are normal messages, attacks cannot be detected by message authentication code (MAC) verification and message decryption. EDDK incorrectly identifies the old message with the changed sequence number as a new message and transmits a false message to the BS. The WSN consumes the energy of the sensor node due to the attack, and confusion occurs because it cannot determine whether the message is false. In this paper, we propose a scheme for detecting retransmission attacks through sequence number encryption. The proposed scheme creates a new sequence key using a pairwise key. The proposed scheme encrypts the sequence number using the sequence key and detects a replay attack occurring in the compromised node. In addition, the proposed scheme performs sequence number verification by selecting an intermediate verification node to save the energy of sensor nodes. The proposed scheme is expected to enhance the security of the WSN and improve energy efficiency in areas with many attacks.

This paper is composed as follows. Section 2 describes the WSN, network layer attack, and EDDK. Section 3 describes the proposed scheme. Section 4 shows the verification of the proposed technique through a simulation. The last section includes our conclusions and future research.

II. RELATED WORKS

A. Wireless sensor networks (WSNs)

WSNs consist of low-power sensor nodes equipped with one or more sensors, processors, memory, actuators, etc., as well as base stations that analyze and transmit collected information to users. Since sensor nodes are inexpensive, dozens to thousands of sensor nodes are used in large areas. The deployed sensor node detects changes in information, such as sound, vibration, and temperature. A sensor node transmits collected information to a BS using wireless communication, and the BS analyzes the information collected by the sensor node and informs the user of the result. Since a sensor node transmits information wirelessly without the need for wiring, installation and maintenance costs are reduced. WSNs can be applied to various areas where situations need to be continuously monitored, such as for military surveillance, disaster recovery, exploration of hazardous areas, and earthquake detection. However, sensor nodes have a limited amount of available resources, such as low energy, short communication radius, low computing capacity, and limited memory.

B. Network layer attack

A sensor node has the advantage of being inexpensive, but it has hardware constraints and is vulnerable to physical security because it is located outside and performs wireless communication. An attacker can compromise sensor nodes using physical attacks, such as node replication attacks and clone attacks. An attacker sniffs data using a compromised node and modulates the data being transmitted.

In addition, an attacker can exhaust the energy of a sensor node by continuously sending data transmission requests and network connection creation requests. Finally, a routing attack can be attempted by providing false routing information and manipulating the routing protocol.

An attacker can disrupt routing by changing the routing message or by sending an old message. Types of attacks include bogus routing information attacks, hello flood attacks, replay attacks, and Sybil attacks [8], [9]. These attacks are targeted at a sensor network and disrupt routing and cause intentional errors. Due to these attacks, WSNs delay the transmission of information and the energy of sensor nodes is consumed. Security research using sequence number and time-stamp algorithms has been conducted to defend against various attacks in WSNs [10], [11]. However, the more complex the security schemes, the more energy consumed by sensor nodes. Therefore, in-depth research is being conducted to improve energy efficiency while applying security techniques to WSNs with limited energy.

C. Energy-efficient distributed deterministic key management scheme (EDDK)

EDDK is a protocol focusing on the configuration and maintenance of pairwise keys and local cluster keys. Unlike centralized or location-based key management systems, EDDK has a very flexible scheme because there is no single node sharing the master key with the BS. EDDK does not decrypt all encrypted messages transmitted from the sensor network, even if an attacker compromises one sensor node to obtain the key. EDDK derives a separate encryption key and MAC key using individual keys instead of a master node. This scheme plays a major role in enhancing the security of data transmission. EDDK consists of a key establishment phase, a data transfer phase, and a key maintenance phase. First, we provide a description of the key establishment phase. A sensor node identifies a neighbor node by broadcasting the network subscription message. The neighbor node identified by the sensor node generates a pairwise key using the encryption key. The pairwise keys of sensor nodes a and b are generated according to the equation (1).

$$K_{ab} = f\,(K_a \oplus K_b,\ SN_a \oplus SN_b) \qquad (1)$$

In Equation (1), K_{ab} is the generated pairwise key, which is shared by sensor nodes a and b. SN_a and SN_b are random numbers generated by each sensor node, adding complexity to pairwise key generation. Pairwise key generation must be completed in less than 10 s to be safe from attackers [12]. Therefore, after a specified time, the sensor node secures storage space by deleting individual keys, random numbers, pseudo-random functions, and initial keys shared with neighbor nodes. Even if the attacker compromises the sensor node, the attacker cannot generate the encryption key of another sensor node because they do not know the deleted element. Therefore, EDDK improves security by removing these factors. In EDDK the local cluster key is a key shared by all neighbor nodes and used to protect all local broadcast messages, such as routing control messages. The pairwise key and local cluster key should be updated periodically to maintain WSN security. The key update frequency uses a sequence number. The encryption key has a sequence number, and the key is updated when the sequence number reaches a predetermined threshold. Then, the sequence number is initialized to 0 to calculate the next key update frequency. Through this, the message maintains freshness, and a replay attack can be prevented because a different sequence number is assigned each time. Table 1 shows the neighbor table of a sensor node [5].

Table I: Neighbor table

Type	Size (Bytes)
Sensor Node ID	2
Pairwise Key	8
Pairwise Key Sequence Number	2
Local Cluster Key	8
Local Cluster Key Sequence Number	2

The following is a description of the data transfer phase. When a sensor node receives a packet, it checks the ID and sequence number of the sensor node. In EDDK, encryption is optional, but all messages are authenticated. If the ID of the sensor node exists in the neighbor table and the sequence number matches, the MAC is calculated and compared for verification. After this process, the message is decrypted. Therefore, the sequence number and the ID of the sensor node can be transmitted without encryption and drop wrong messages early. Through this process, EDDK can save energy resources for authentication and message decryption through MAC. Finally, the key maintenance phase is explained. EDDK periodically performs key updates to maintain security, preventing attackers from decrypting previous messages. Also, if compromised nodes are identified, the key is immediately discarded. The network can be maintained by updating pairwise and local cluster keys shared with a compromised node and separating the compromised node. The new node is authenticated using the elliptic curve digital signature algorithm (ECDSA), and then it joins the network [13]. ECDSA is a public key-based algorithm that uses a small elliptic curve cryptography (ECC) key. ECDSA is suitable for use in WSNs with hardware constraints. According to the Diffie-Hellman algorithm, each sensor node can independently calculate the pairwise key using its own private key and the public key of another node [14], [15]. In this way, EDDK joins a new sensor node to the network and shares a pairwise key.

III. REVIEW CRITERIA

A. Motive

EDDK saves energy for MAC authentication and message decryption through early verification. For early verification, EDDK transmits the ID and sequence number of a sensor node together with the message, without encryption. The sequence number is used to defend against replay attacks and ensures the freshness of the message. However, an attacker can verify the unencrypted sequence number by sniffing data through compromised nodes. The attacker can then attempt a replay attack by changing the sequence number. In the message modulated by the attacker, the ID and sequence number of the sensor node match.

Therefore, MAC authentication and message decoding are performed. Since the replay attack retransmits the previous message, the false message is judged as a normal message, even after the verification procedure is performed through MAC.

The false message, whose message verification has been completed, is transmitted to the BS. In addition, the attacker can speed up the initialization of the key by modulating the sequence number. A sensor node that has undergone a replay attack consumes energy through unnecessary transmission, reception, and verification. Additionally, the replay attack accelerates the key update frequency, reducing the overall energy of the sensor network.

B. Assumption

The sensor nodes are randomly placed, and the BS is placed at the bottom right. There is no sensor node that is completely depleted of energy. The energy of the sensor node is randomly set in a state not exceeding a maximum of 1 J. When the energy of the sensor node is exhausted, it is rerouted using the sensor node of the neighboring node. The BS is not attacked by the security system, and the position and energy status of all sensor nodes are periodically reported by the sensor nodes.

C. Detailed proposed scheme

In this paper, we propose a scheme to protect against replay attacks that modulate sequence numbers by encrypting sequence numbers. In addition, the proposed scheme is expected to improve energy efficiency through filtering of false messages. EDDK reduces the unnecessary energy consumption caused by decoding wrong and duplicate messages by early verification; this is done by using the sequence number and the sensor node ID only. Therefore, the unencrypted sensor node ID and sequence number are included in the message and transmitted. An attacker can obtain the sequence number by sniffing the data transmitted through a compromised node. The attacker can then attempt a replay attack that manipulates the sequence number using the compromised node. When this happens, after confirming the sequence number and sensor node ID, EDDK judges the message as normal and proceeds with MAC verification and message decryption. Since the false message is the same as the previous message, except for the sequence number, MAC verification and message decoding are determined to be normal. The sensor node that received the false message re-encrypts the message with a pairwise key for verification with another sensor node and attaches the new sequence number and transmits it. The false message is transmitted to the BS through the sensor node. The BS may cause confusion about the received information because messages containing the same content are continuously transmitted. The proposed scheme prevents replay attacks that change the sequence number through sequence number encryption. The sequence number used in EDDK was used as an early verification factor for previous messages. However, if an attacker attempts a replay attack with a sequence number change, it is difficult to guarantee the integrity of the message. The proposed scheme detects replay attacks through sequence number encryption and maintains the security of the sensor network. The sequence number encryption method in the proposed scheme is as follows. First, the sensor node generates pairwise keys and local cluster keys of neighbor nodes in the same way as EDDK. The proposed scheme creates a new sequence key using a pairwise key and initial key. The sequence key shared by sensor nodes c and d is generated according to equation (2).

$$K_{cd} = f\,(k_{pk(c,d)}, (ID_c + ID_{\,d}) \oplus K_I) \qquad (2)$$

K_{cd} is a sequence key and $k_{pk(c,d)}$ is a pairwise key shared by sensor nodes c and d. K_I is the initial key, which is set before deployment. Since the elements necessary to generate a sequence key have been exchanged in advance, the sequence key can generate a key without additional message exchange. Figure 2 shows a flowchart for sequence key updates.

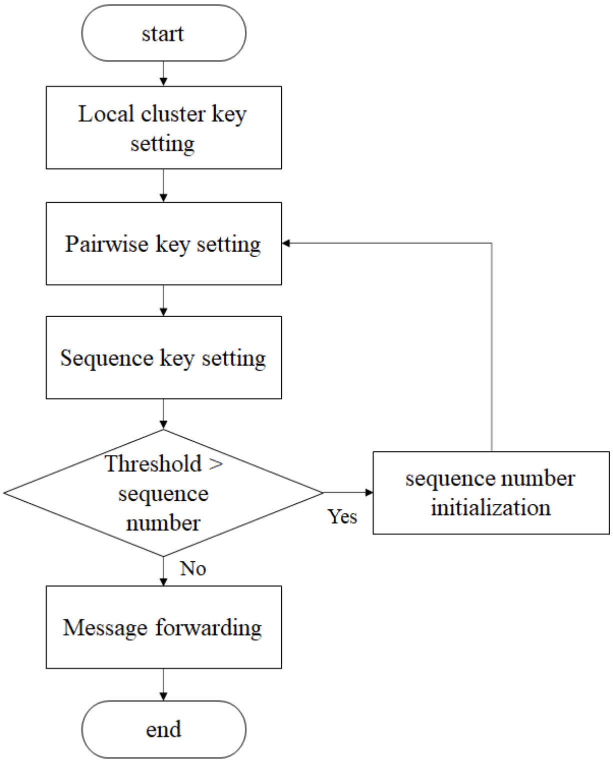
start
Local cluster key setting
Pairwise key setting
Sequence key setting
Threshold > sequence number
Yes
sequence number initialization
No
Message forwarding
end

Fig. 2. Security Key Update Process

The pairwise key is updated when the sequence number reaches the threshold. The sequence key is updated using the newly updated pairwise key. Therefore, since the sequence key update frequency and the pairwise key update frequency are the same, the proposed scheme does not require additional elements for setting the sequence key update frequency. The event content is encrypted with the pairwise key, and the sequence number is encrypted with the sequence key. The proposed scheme stores two ciphertexts separately when generating a message.

When a false message is determined early by decoding only the sequence number, the sensor node can save energy related to decoding the message. However, if the sequence number is checked every time in the proposed scheme, energy consumption for decoding occurs; this method consumes more energy than EDDK. Therefore, the proposed scheme selects an intermediate verification node.

The proposed scheme can save energy because the message is verified by decoding the sequence number only at the intermediate verification node. In the proposed scheme, the sensor nodes between the source node that detects the event and the selected intermediate node go through the sequence number verification procedure each time using their sequence key.

When the message reaches the intermediate verification node, the sequence number is encrypted using a sequence key shared between the intermediate verification nodes. Figure 3 shows the intermediate verification node used in the proposed scheme.

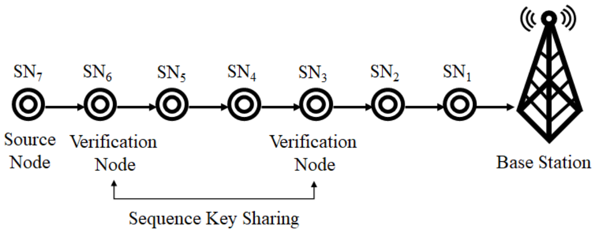

Fig. 3. Message Transmission Using an Intermediate Verification Node

If the proposed scheme is used in an area where many attacks occur, early verification through the sequence number is possible only when the message reaches the intermediate verification node. Since the timing of filtering for false messages is delayed, more energy may be consumed by the sensor network. In the opposite situation, the energy efficiency of the sensor network is improved by reducing unnecessary decoding energy when using the proposed scheme.

IV. EXPERIMENTAL RESULTS

The proposed scheme simulated the process of setting the pairwise key and local cluster key and transmitting an event message after the sensor node was deployed. The proposed scheme verified the security against replay attacks through simulation, and the initialization count and energy efficiency were compared with those of EDDK. Table 2 shows the parameters and values used in the simulation environment.

Table II: Experiment Parameters

Simulation Parameter	Value
Sensor Field Size	600 m x 200 m
Sensor Node Type	MICAz
Number of Sensor Node	300
Maximum Sensor Node Energy	1 J
Radio Range	150 m
Base Station Location	(x, y) = (600 m, 200 m)
Data Packet Size	36 bytes
MAC Size	1 byte

In the experiment, an event occurred 5000 times, and normal and false messages are randomly generated and transmitted according to the attack ratio. The sensor node was designed according to the specifications of the MICAz model. The energy of the sensor node was randomly selected to verify various environments, and the maximum energy that the sensor node can have is 1 J. If the residual energy of the sensor node is less than a preset value, the sensor node cannot monitor the situation. Since the sensor node of the proposed scheme is designed with reference to the MICAz model, the energy consumed during calculation and packet transmission is the same as the energy consumed by the actual sensor node. The designed sensor node consumes 3.5 nJ of energy per clock for calculations. The sensor node consumes 9.2 nJ of energy per clock in the active state and 3 pJ of energy per clock in the sleep state. In addition, the sensor node consumes 0.6 μJ of energy when transmitting a one-bit packet. Conversely, when receiving a one-bit packet, it consumes 0.67 μJ of energy [16]. The proposed scheme uses the CBC-MAC used in EDDK. CBC-MAC is generated using RC5 (round 12) [17]. The energy consumed to produce CBC-MAC is 49.92 μJ per cycle [18].

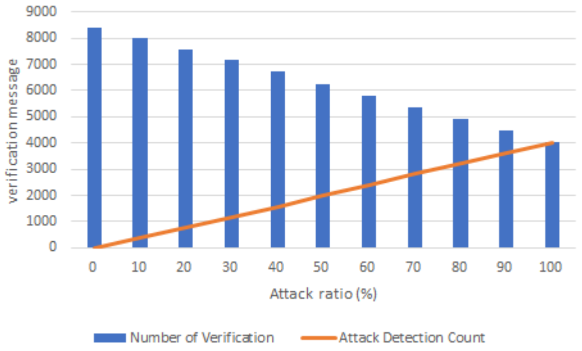

Fig. 4. New Types of Replay Attack Detection

Figure 4 shows the number of message verifications and the replay attack detection count according to the attack ratio in the proposed scheme. The proposed scheme consumes a lot of energy if there are no attacks in the networks since all sensor nodes in the path from the source node to the BS must be verified. Conversely, the proposed scheme can reduce the energy consumption of sensor nodes by filtering false messages through early verification when there are many attacks. When using the proposed scheme, the graph shows that a replay attack that changes the sequence number is detected. However, the existing scheme does not encrypt the sequence number; therefore, it cannot detect a new type of replay attack.

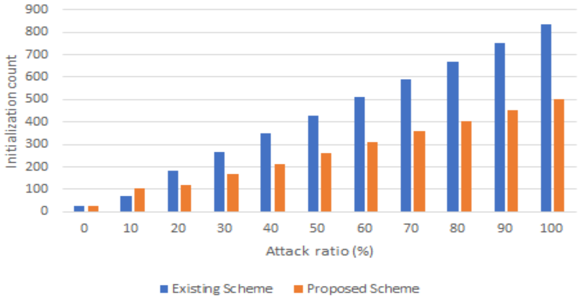

Fig. 5. Initialization Count According to the Attack Ratio

Figure 5 shows the key initialization count for the existing scheme and the proposed scheme. The proposed scheme and EDDK initialize the corresponding key when detecting an attack; therefore, more key initialization instances occur when there are many attacks. However, EDDK cannot detect a new type of replay attack that changes the sequence number.

Therefore, the attacker manipulates the sequence number so that many instances of key initialization occur. For this reason, EDDK executes key initialization more than it does when sending and receiving messages normally. Since the proposed scheme detects a new type of replay attack, additional key initialization due to the attack does not occur. Key initialization of the proposed scheme occurs according to the rules established in EDDK.

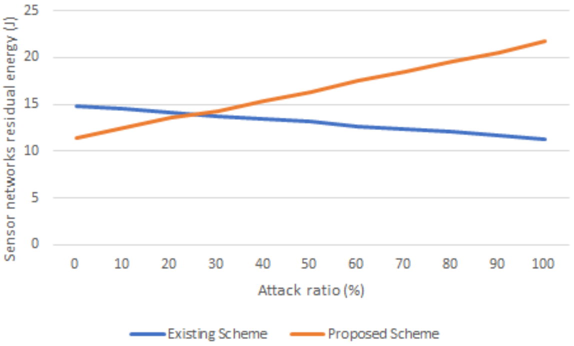

Fig. 6. Sensor Network Residual Energy According to the Attack Ratio

Figure 6 shows the residual energy of the sensor network according to the attack ratio for the existing scheme and the proposed scheme. Both schemes have the same number of message verifications from the source node to the BS in the absence of attacks; thus, the energy consumed in the verification procedure is the same. However, in the proposed scheme, additional sequence key setting and sequence number decoding are performed. Therefore, the proposed scheme consumes more energy than the existing scheme, and the existing scheme shows better energy efficiency in environments where the attack ratio is less than 30%. Conversely, the proposed scheme has better energy efficiency in an environment in which replay attacks occur frequently. In an environment with an attack ratio of 100%, the energy efficiency of the proposed scheme is improved by about 7.112% relative to the existing scheme.

V. CONCLUSIONS

WSNs enable users to acquire situational information using various sensors at desired locations. For this reason, WSNs are used in many fields that require real-time monitoring. However, the sensor nodes used in WSNs have many restrictions and are easily compromised because they are placed outside. An attacker can attempt various attacks that damage the network by using compromised nodes. These types of attacks include hello flooding attacks, selective forwarding attacks, and sybil attacks. Many security techniques have been researched to defend against these attacks, including EDDK, which is a scheme that can be used to defend against various attacks through key management. The EDDK scheme protects messages using a pairwise key and a local cluster key against attacks occurring at the network layer. EDDK is more energy efficient than other security schemes and enhances security through periodic security key updates. The sequence number of EDDK is used to prevent replay attacks and improve energy efficiency; however, it also represents a new attack target for attackers. If an attacker modulates the sequence number, a new type of replay attack is possible. This replay attack consumes energy of sensor nodes by sending the previous message. This also causes confusion in the information acquired by the user. Therefore, we propose a scheme to encrypt the sequence number to prevent this new type of replay attack. In addition, the proposed scheme performs sequence number verification by selecting an intermediate verification node to increase the energy efficiency. The proposed scheme detects replay attacks that modulate sequence numbers through experiments and improves the energy efficiency by about 7% in areas with high attack ratios. In the future, in order to further increase energy efficiency, we plan to conduct research on efficient selection of intermediate verification nodes according to the hop count.

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. Suzuki, Makoto, et al. "A high-density earthquake monitoring system using wireless sensor networks." Proceedings of the 5th international conference on Embedded networked sensor systems, pp.373-374, 2007.
3. Y. Zhou, Y. Fang, and Y. Zhang, "Securing wireless sensor networks: a survey," IEEE Communications Surveys & Tutorials, vol. 10, no. 3, pp. 6–28, 2008.
4. Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," IEEE Communications Surveys & Tutorials, vol. 8, no. 2, pp. 2–22, 2006.
5. 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.
6. 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.
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. Karlof, Chris, and David Wagner. "Secure routing in wireless sensor networks: Attacks and countermeasures." Ad hoc networks vol. 1, no. 2-3, pp. 293-315, 2003.
9. Kavitha, T., and D. Sridharan. "Security vulnerabilities in wireless sensor networks: A survey." Journal of information Assurance and Security vol. 5, no. 1, pp. 31-44, 2010.
10. Ping, Su. "Delay measurement time synchronization for wireless sensor networks." Intel Research Berkeley Lab 6, 1-10, 2003.
11. Medjadba, Yasmine, and Somia Sahraoui. "Intrusion detection system to overcome a novel form of replay attack (data replay) in wireless sensor networks." International Journal of Computer Network and Information Security vol. 8, no. 7 pp. 50, 2016.
12. 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.
13. S. Vanstone, "Responses to NIST's proposal," CACM, vol. 35, no. 7, pp. 50–52, 1992.

14. W. Diffie and M. E. Hellman, “New directions in cryptography,” IEEE Transactions on Information Theory, vol. 22, no. 6, pp. 644–654, 1976.
15. De Meulenaer, Giacomo, 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.
16. 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.
17. Kukkurainen, Juha, Mikael Soini, and Lauri Sydanheimo. "RC5-based security in wireless sensor networks: Utilization and performance." WSEAS Trans. Comput vol. 9, no. 10, pp. 1191-1200, 2010.
18. 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 Computing, Sungkyunkwan University, Korea.