Kim and Song *EURASIP Journal on Wireless Communications and Networking* (2019) 2019:155
https://doi.org/10.1186/s13638-019-1470-9

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Energy-efficient and secure mobile node reauthentication scheme for mobile wireless sensor networks

BoSung Kim* and JooSeok Song

Abstract

Mobile wireless sensor networks (MWSNs) are a relatively new type of WSN where the sensor nodes are mobile. Compared to static WSNs, MWSNs provide many advantages, but their mobility introduces the problem of frequent reauthentication. Several mobile node reauthentication schemes based on symmetric key cryptography have been proposed to efficiently handle frequent reauthentication. However, due to security weaknesses such as unconditional forwarding or low-compromise resilience, these schemes do not satisfy the security requirements of MWSNs. In this paper, we propose an energy-efficient and secure mobile node reauthentication scheme (ESMR) for MWSNs that satisfies the security requirements of MWSNs by addressing the security weaknesses of previous studies. ESMR prevents unconditional forwarding by allowing a foreign cluster head to authenticate mobile nodes, providing high-compromise resilience because it limits the use of cryptographic keys for different purposes. Security analysis shows that ESMR meets security requirements and can prevent relevant security attacks. Performance evaluation shows that ESMR is suitable for multi-hop communication environment, where the number of hops between the mobile node and cluster head is two or more. Specifically, ESMR requires less than 6% increased total energy consumption and 3% increased reauthentication latency compared with previous studies; hence, it introduces negligible performance overhead. Considering both performance and security aspects, ESMR also can be applied to single-hop communication environment.

Keywords: Wireless sensor networks, Mobile wireless sensor networks, Mobile sensor node, Security, Authentication, Key agreement, Multi-hop communication

1 Introduction

With the development of the internet of things (IoT), wireless sensor networks (WSNs) are attracting significant attention as a fundamental technology in the IoT. WSNs consist of many sensor nodes that collect data from their surrounding environments and send them to a base station in a multi-hop fashion. Mobile WSNs (MWSNs) are a new type of WSN in which sensor nodes are mobile. By supporting mobility, they can achieve better network performance than traditional static WSNs. Recent studies [1, 2] have shown that MWSNs not only extend network lifetime, but also improve connectivity and coverage. MWSNs also support more types of applications than static WSNs in the following fields: patient monitoring [3], animal monitoring and tracking [4], and object monitoring [5].

Although MWSNs provide many advantages compared to static WSNs, the underlying sensor mobility introduces the problem of frequent reauthentication. Many MWSN applications, such as battlefield surveillance, habitat monitoring, and healthcare, require secure communications, and authentication is an essential first step for secure communication. In MWSNs, sensor nodes continuously change their positions, causing frequent communication link and network topology changes [6, 7]. Consequently, authentication is repeatedly required to establish secure communications whenever the communication link changes. Such frequent reauthentication can cause significant energy consumption for resource-constrained sensor nodes, and it is important to efficiently handle

*Correspondence: bokor@yonsei.ac.kr
Department of Computer Science, Yonsei University, 50 Yonsei-ro, Seodaemun-Gu, Seoul, Republic of Korea

communication and computation overheads. Although there have been many studies to support mobility in existing internet services, they are not applicable to MWSNs due to resource constraints of the sensor node [8]. Therefore, lightweight security mechanisms that explicitly consider sensor mobility are required to handle frequent reauthentication in MWSNs.

Several mobile node reauthentication schemes for MWSNs have been proposed to efficiently handle frequent reauthentication based on symmetric key cryptography [9–11], which can provide lightweight reauthentication mechanisms suitable for resource-constrained sensor nodes. However, these schemes generally have significant security weaknesses.

In most applications, the sensor node is placed in a location easily accessible by an adversary; hence, the adversary is more likely to be able capture the sensor node and extract cryptographic secrets. Therefore, any mobile node reauthentication scheme should provide high-compromise resilience, where the compromised node reveals no information about links it is not directly involved with. However, Han et al.'s schemes [9, 10] do not provide high-compromise resilience. If even a single-cluster head is captured, other nodes that do not share pairwise keys with the compromised cluster head can be affected, i.e., an adversary can compromise pairwise keys of other nodes that are not directly involved with the compromised cluster head.

Adversaries can also eavesdrop on the radio frequencies of MWSNs due to the open nature of wireless communication, and alter or spoof messages to launch denial of service (DoS) attacks or drain receiver resources. Authentication to provide the assurance of identity of the communicating node is essential to prevent such attacks. However, Jiang et al.'s scheme [11] allows a foreign cluster head to unconditionally forward mobile nodes' reauthentication requests to the home cluster head without preliminary verification, which could be used to launch DoS attacks on the home cluster head.

In this paper, we propose an energy-efficient and secure mobile node reauthentication scheme (ESMR) for MWSNs. ESMR focuses on satisfying the security requirements of MWSNs by addressing the security weaknesses of the existing mobile node reauthentication schemes [9–11]. ESMR uses two additional keys (key derivation key (KDK) and authentication key (AK)) in addition to the pairwise key to prevent unconditional forwarding. ESMR also provides high -compromise resilience because it limits the use of the KDK and AK for different purposes. Each cluster head has its own KDK that it shares as a group key with neighboring cluster heads. During initial authentication, the home cluster head generates an AK for a mobile node for the next authentication by using its KDK. When the mobile node moves to a new location and initiates the reauthentication procedure with a foreign cluster head, the foreign cluster head can authenticate the mobile node by using the AK. Consequently, unconditional forwarding is prevented. The KDK is only used to generate AKs, and AKs are only used for authenticating mobile nodes. Thus, the compromised cluster head does not affect other nodes that do not share pairwise keys with the compromised cluster head. Security analysis shows that by addressing the security weaknesses of the existing schemes, ESMR meets security requirements of MWSNs and can prevent relevant security attacks. The main contributions of this work can be described as follows:

- We show the security weaknesses of existing mobile node reauthentication schemes based on symmetric key cryptography [9–11].
- We propose ESMR, which satisfies the security requirements of MWSNs by addressing the security weaknesses of existing mobile node reauthentication schemes.
- The security of ESMR is formally verified by using the automated validation of internet security protocols and applications (AVISPA) tool [12].
- Simulations are conducted using OMNeT++ with the INET framework 3.6.3 to evaluate the performance of ESMR in terms of energy consumption and reauthentication latency.

The remainder of this paper is organized as follows. Section 2 discusses related works and their problems. Section 3 briefly reviews existing mobile node reauthentication schemes [9–11] and discusses identified security weaknesses. Section 4 provides an overview of ESMR, and Section 5 provides the details. Section 6 analyzes the security of ESMR. Section 7 evaluates the performance of ESMR. Finally, Section 8 summarizes and concludes the paper.

2 Related works

Mobile node reauthentication schemes for MWSNs can be classified into two types according to their encryption techniques: symmetric key cryptography schemes [9–11, 13–15] and public key cryptography schemes [16–19]. Symmetric key cryptography schemes can be further classified into two types: post-deployment key establishment schemes [9–11, 13] and hybrid schemes that use both random key pre-distribution and post-deployment key establishment mechanisms [14, 15].

2.1 Symmetric key cryptography schemes

Han et al. proposed two ticket-based schemes [9, 10]. In [9], each cluster head has its own ticket generation key (TGK) that is used to generate tickets, which is shared as a group key with neighboring cluster heads. When a mobile node moves to a new location and initiates the reauthentication procedure, the mobile node and foreign cluster head authenticate each other and establish a pairwise key using the ticket. However, this scheme may not work properly in situations where sensor nodes are irregularly distributed and mobile nodes move between non-neighboring cluster heads.

Therefore, an improved ticket-based scheme was proposed [10] that utilizes the neighboring cluster head list (NCL) for reauthentication. Two non-neighboring cluster heads compare their NCLs to find a common neighboring cluster head. Then reauthentication can be performed through the common neighboring cluster head even if two cluster heads are not neighbors.

The main disadvantage of these schemes is that communication overhead is concentrated in mobile nodes, because the mobile nodes directly transfer the information required for reauthentication, e.g., tickets and NCLs. The schemes also do not provide high-compromise resilience, since each cluster head shares its TGK as a group key with neighboring cluster heads. Therefore, multiple TGKs are exposed if even a single-cluster head is captured. An adversary can then obtain secret information included in all generated tickets using the exposed TGKs and compromise multiple nodes' communication security using secret information obtained from the tickets.

In [13], Bilal and Kang proposed a ticket-based authentication suite that supports multiple secure connections. The authentication suite consists of two mobile node reauthentication protocol, SRP1, and SRP2, which support multiple secure connections between the mobile node and multiple cluster heads. In SRP1 and SRP2, the mobile node and foreign cluster head can authenticate each other directly using the ticket. However, SRP1 and SRP2 have the same disadvantage as [9] and [10], in which the communication overhead is concentrated in the mobile node. Moreover, since SRP2 requires the involvement of the base station, there is a problem that the reauthentication latency is long.

To reduce the communication overhead of mobile nodes, Jiang et al. [11] proposed a mobile node reauthentication scheme without using tickets. In the scheme, a foreign cluster head forwards the reauthentication request of a mobile node to the home cluster head. The home cluster head then authenticates the request on behalf of the foreign cluster head. Since the information required for reauthentication is exchanged between the foreign and home cluster head, the communication overhead in mobile nodes is reduced. However, the scheme has a security weakness called unconditional forwarding. Since there are no shared secrets between the foreign cluster head and mobile node, foreign cluster heads cannot authenticate reauthentication requests of mobile nodes and unconditionally forward reauthentication requests to the home cluster head. This unconditional forwarding can be lead to DoS attacks on the home cluster head.

The previous schemes use only the post-deployment key establishment mechanism, whereas hybrid schemes [14, 15] use both random key pre-distribution and post-deployment key establishment mechanisms. In these schemes, by default, two nodes authenticate each other and establish a pairwise key based on existing random key pre-distribution schemes [20, 21]. For random key pre-distribution schemes, a set of keys are randomly chosen from a large key pool and pre-stored in each sensor node. During the key discovery phase, two nodes exchange pre-stored key identifiers to find a common key. They use the post-deployment key establishment scheme if two nodes do not share a common key to establish a pairwise key with the help of the base station.

The main disadvantage of hybrid scheme is that they require a minimum network density for the random key pre-distribution mechanism. Although two nodes that do not share a common key can establish a pairwise key using the post-deployment key scheme, multi-hop communication with the base station incurs longer communication delay and consumes more energy as hop distance between the mobile node and base station increases.

2.2 Public key cryptography schemes

In [16], Gandino et al. proposed an authentication and key establishment scheme based on public key cryptography for mobile and static WSNs. In the scheme, authentication tables are used to reduce communication and computational overhead due to the verification of digital certificate. The authentication table stores information necessary for a node to authenticate the other nodes in the network and is distributed to each node before deployment. In the key establishment phase, two nodes can authenticate each other's public keys directly using the authentication table instead of the digital certificate. However, there is a problem that the sensor node generates a pairwise key by performing public key encryption and decryption operations which cause a high-computation overhead. Especially, the computation overhead of the mobile node becomes more severe because the mobile node generates a new pairwise key every time reauthentication is performed.

With the development of elliptic curve cryptography (ECC) optimization techniques, several ECC-based mobile node reauthentication schemes have been proposed [17–19]. Zhang et al. [17] used the elliptic curve digital signature algorithm and elliptic curve Diffie-Hellman key agreement to dynamically generate pairwise keys. However, the scheme is not suitable for large-scale WSNs because of the overhead required for certificate management.

Seo et al. [18] proposed an ECC-based scheme without certificates to overcome this limitation using pairing-free certificateless hybrid signcryption scheme (CL-HSC) to dynamically provide both mobile node authentication and key agreement. The properties of the CL-HSC ensure that a pairwise key can be generated without requiring expensive pairing operations or certificate exchange. However, the scheme still requires expensive ECC point multiplications for mobile nodes to generate long-term pairwise keys, i.e., mobile nodes must perform multiple expensive ECC multiplications repeatedly whenever they move and are connected to a new cluster head.

Omar et al. [19] proposed a trusted third party based mobile node reauthentication scheme using ECC. In the scheme, when a mobile node wants to join a new cluster, it sends a join request message, including its public key, to a foreign cluster head. Upon receiving the message, the foreign cluster head requests the base station, which is a trusted third party, to authenticate the mobile node. Since the base station performs verification of the public key, there is no computation overhead of the sensor node due to the expensive ECC multiplication. However, the longer the hops distance between the cluster head and the base station, the longer the re-authentication latency becomes.

3 Analysis of post-deployment schemes

This section briefly reviews existing schemes [9–11] and highlights their security weaknesses.

Han et al.'s schemes [9, 10] use tickets for mobile node reauthentication, and have the same security weakness because they have almost the same reauthentication process. We focus on the mobile node reauthentication process based on [9]. Each cluster head has its own ticket generation key (TGK) that it shares with neighboring cluster heads. In the initial authentication, the home cluster head CH_A generates ticket T for the next reauthentication as follows using its ticket generation key TGK_{CH_A}:

$$\begin{aligned} T &= (t, w) \\ t &= E(K_{TGK_{CH_A}}, TS||R_1||K_{MN-CH_A}) \\ w &= MAC(K_{TGK_{CH_A}}, ID_{MN}||t) \end{aligned} \tag{1}$$

where R_1 is a random number generated by the mobile node MN and K_{MN-CH_A} is a pairwise key between MN and CH_A.

When MN moves to a new location and receives the HELLO message from the foreign cluster head CH_B, it launches the reauthentication process by sending following reauthentication request to CH_B:

$$\begin{aligned} & MN \rightarrow CH_B : ID_{MN}||ID_{CH_B}||t||w||v_1 \\ & v_1 = MAC(K_{M-CH_A}, ID_{MN}||ID_{CH_B}||t||w||v_0) \end{aligned} \tag{2}$$

Since CH_B is a neighboring cluster head of CH_A, it also has the TGK of CH_A. Therefore, CH_B can verify w, and obtain R_1 and K_{MN-CH_A} by decrypting t. CH_B then authenticates MN by verifying v_1 using K_{MN-CH_A} and generates a pairwise key with MN as follows:

$$K_{MN-CH_B} = KDF(R_1||R_0) \tag{3}$$

where R_0 is a random number generated by CH_B.

CH_B finally generates a new ticket T', in the same way that CH_A did in (1) using its TGK and sends following message to MN:

$$\begin{aligned} & CH_B \rightarrow MN : ID_{CH_B}||ID_{MN}||u_3||v_3 \\ & u_3 = E(K_{MN-CH_A}, R_0||v_2||T') \\ & v_2 = H(K_{MN-CH_B}||R_0) \\ & v_3 = MAC(K_{MN-CH_A}, ID_{CH_B}||ID_{MN}||u_3) \end{aligned} \tag{4}$$

Upon receiving the message from CH_B, MN verifies v_3 and obtains R_0 using K_{MN-CH_A}. MN then generates K_{MN-CH_B} in the same way that CH_B did in (3) and finally authenticates CH_B by verifying v_2 using K_{MN-CH_B}.

Assume that cluster head CH_C is another neighboring cluster head of CH_A, but CH_C and CH_B are not neighbors. If an adversary captures CH_C and extracts TGK_{CH_A} from it, the adversary can obtain R_1 and K_{MN-CH_A} by decrypting t included in T using TGK_{CH_A}, as shown in (1). The adversary can then decrypt u_3, included in the message in (4), using K_{MN-CH_A} to obtain R_0. The adversary can finally compromise K_{MN-CH_B} by directly generating K_{MN-CH_B} using R_1 and R_0, as shown in (1). Consequently, the adversary can compromise communication security between MN and CH_B not directly involved with the compromised cluster head. This problem is more serious because each cluster head has multiple TGKs, including TGKs of the neighboring cluster head and its own TGK. Thus, an adversary can compromise communication security for a number of nodes by compromising a single-cluster head.

On the other hand, Jiang et al.'s scheme [11] does not use tickets to reduce communication overhead of the mobile node. In the scheme, reauthentication process is proceeded with the help of the home cluster head, CH_A, rather than using a ticket. When MN wants to launch the reauthentication process, it sends following reauthentication request to the foreign cluster head CH_B:

$$\begin{aligned} & MN \rightarrow CH_B : ID_{MN}||ID_{CH_A}||t_1||MAC_1 \\ & MAC_1 = MAC(K_{MN-CH_A}, ID_{MN}||t_1||H(I)) \end{aligned} \tag{5}$$

where ID_{CH_A} is the identity of CH_A, t_1 is a timestamp, K_{MN-CH_A} is a pairwise key between MN and CH_A, and $H(I)$ is a hashed random number I shared between MN and CH_A.

Upon receiving the reauthentication request from MN, CH_B checks ID_{CH_A} included in the request and finds that the home cluster head of MN is CH_A. CH_B then forwards the reauthentication request of MN to CH_A:

$$\begin{aligned} & CH_B \rightarrow CH_A : ID_{MN}||t_2||t_1||MAC_1||MAC_2 \\ & MAC_2 = MAC(K_{CH_B-CH_A}, ID_{MN}||t_2||t_1||MAC_1) \end{aligned} \tag{6}$$

where t_2 is a timestamp and $K_{CH_B-CH_A}$ is a pairwise key between CH_B and CH_A. After receiving the message from CH_B, CH_A verifies MAC_1 using K_{MN-CH_A} and $H(I)$ and authenticates MN on behalf of CH_B.

In the above scheme, CH_B cannot verify validity of the reauthentication request of MN in (5), because there is no shared secret between CH_B and MN. Consequently, CH_B unconditionally forwards the reauthentication request of MN as shown in (6). This unconditional forwarding allows DoS attacks on CH_A through neighboring cluster heads of CH_A, because an adversary can easily alter or spoof eavesdropped messages in MWSNs.

4 Overview of ESMR

In this paper, we propose an energy-efficient and secure mobile node reauthentication scheme (ESMR) for MWSNs. ESMR supports mutual authentication and key establishment between mobile nodes and cluster heads. ESMR is based on Jiang et al.'s scheme [11], which has the lowest computation and communication overheads for mobile nodes among existing schemes, and falls into the post-deployment key establishment scheme category. Figure 1 presents an overview of ESMR.

Each cluster head in ESMR has its own key derivation key (KDK) to generate the authentication keys (AKs) for mobile nodes. Prior to network operation, each cluster head shares its KDK as a group key with neighboring cluster heads (phase 0). During initial authentication (phase 1), the home cluster head CH_A, that a mobile node MN first connects after deployment, generates an AK for the mobile node by using its KDK for the next authentication. MN initiates the reauthentication procedure, when it moves to a new location and connects to a foreign cluster head CH_B. During reauthentication (phase 2), CH_B generates an AK using the KDK of CH_A and authenticates MN.

ESMR addresses the security weaknesses of schemes proposed by Han et al. [9, 10] and Jiang et al. [11]. First, ESMR prevents unconditional forwarding by allowing foreign cluster heads to authenticate mobile nodes. Second, ESMR provides high-compromise resilience by limiting the use of the KDK and AK for different purposes: the KDK is only used to generate AKs, and AKs are only used for authenticating mobile nodes. Table 1 provides the major notations used in this paper.

4.1 Network model

Following mobility-aware medium access control protocols for WSNs [22, 23], we consider a heterogeneous sensor network, consisting of a base station, cluster heads, and sensor nodes, as shown in Fig. 2. There are N_1 and N_2 cluster heads and sensor nodes, respectively, with where $N_1 << N_2$, and hence the total number of nodes in the network = $N_1 + N_2$.

Cluster heads are high-end sensor nodes with more resources in terms of computational power, storage, and battery life than the sensor nodes. The communication range of a cluster head is also larger than that of a sensor node. Cluster heads compose a stationary backbone network and periodically broadcast lightweight beacon messages to inform nodes of their presence. Considering the wide communication range of cluster heads, we assume that a foreign cluster head is a neighboring cluster head to the home cluster head.

Sensor nodes act as cluster members and can be stationary or mobile. A sensor node initiates the reauthentication procedure with a foreign cluster head when it moves to a new location and receives a beacon message from the foreign cluster head. Since the communication range of the sensor node is smaller than that of a cluster head, stationary nodes relay mobile node messages to cluster heads.

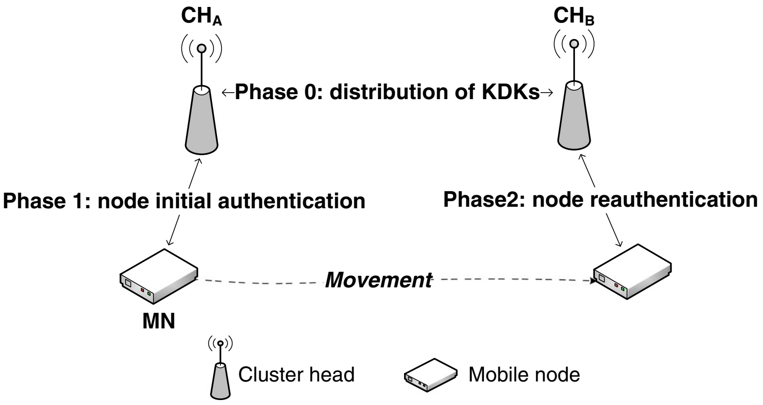

CH_A
CH_B
←Phase 0: distribution of KDKs→
Phase 1: node initial authentication
Phase2: node reauthentication
Movement
MN
Cluster head
Mobile node

Fig. 1 Overview of ESMR

Table 1 Notations

Symbol	Description
ID_A	Identity of node *A*
TS_i	Timestamp
K_{A-B}	Pairwise key between node *A* and node *B*
KDK_A	KDK of node *A*
AK_A	AK of node *A*
$E(K, m)$	Encrypt message *m* with key *K*
$MAC(K, m)$	Message authentication code of message *m* with key *K*
$\|\|$	Concatenation
$H(\cdot)$	Hash function
f	Pseudorandom function
$\oplus$	Exclusive-OR operation

4.2 Adversary model

We assume that mobile nodes and cluster heads can be attacked passively or actively. Because of the open nature of wireless communication channels, an adversary can easily perform passive attacks, such as eavesdropping and traffic analysis, to gather information without being detected. In active attacks, an adversary may inject, intercept, or replay messages to disrupt network functionality or degrade network performance. We also assume that mobile nodes and cluster heads can be captured by an adversary. Because of the unattended nature of WSNs, nodes can also be physically captured by an adversary. Once a node is captured, all its stored secret information can be revealed to an adversary. An adversary can then utilize this secret information to perform the impersonation attack or compromise communication security. General security mechanisms, such as authentication and encryption, cannot prevent such insider attacks. However, secure reauthentication schemes should minimize insider attacks.

5 Details of ESMR

ESMR consists of three phases: the distribution of KDKs, node initial authentication, and node reauthentication. For simplicity of explanation, we describe ESMR based on Fig. 1.

5.1 Phase 0: distribution of KDKs

Each cluster head has its own KDK to generate AKs for mobile nodes. Prior to network operation, each cluster head shares its KDK as a group key with neighboring cluster heads. We assume that each cluster head has established pairwise keys with neighboring cluster heads after deployment. This can be accomplished with the help of the base station in a similar way to Han et al.’s scheme [9]. Each cluster head then uses the pairwise key to securely distribute its KDK to neighboring cluster heads. For example, when CH_A wants to share KDK_{CH_A} with CH_B, it computes e_0 and v_0, and then sends the following message to CH_B with current timestamp TS_0:

$$CH_A \rightarrow CH_B : ID_{CH_A}||ID_{CH_B}||TS_0||e_0||v_0$$
$$e_0 = E(K_{CH_A-CH_B}, KDK_{CH_A})$$
$$v_0 = MAC(K_{CH_A-CH_B}, ID_{CH_A}||ID_{CH_B}||TS_0||e_0)$$

The reason for sharing the KDK as a group key is that it is difficult to predict the movement of mobile node. To use a pairwise key between two cluster heads to generate AKs, we have to accurately predict the network cluster head the mobile node will connect to after movement. However, existing movement estimation techniques are inaccurate or require additional special hardware [24]. For example, although calculating an angle of arrival incurs no additional costs, is error prone, and energy inefficient. On the other hand, although global positioning systems can measure precise and absolute coordinates, they require additional expensive hardware.

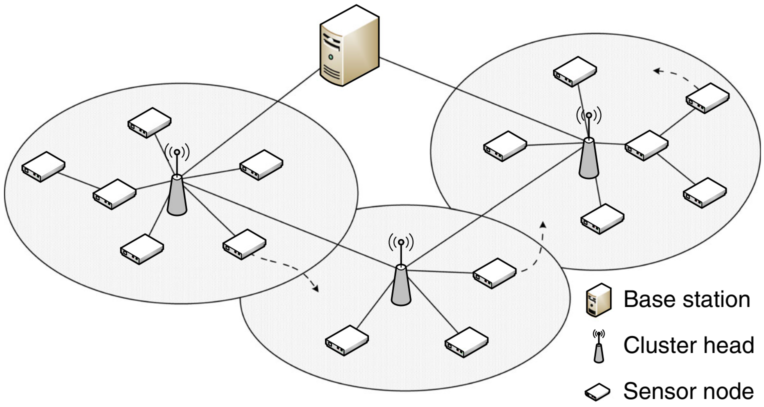


Fig. 2 Heterogeneous sensor network

5.2 Phase 1: node initial authentication

Initial authentication is performed when MN joins the network for the first time after deployment. Let CH_A be the home cluster head that MN first connects to after deployment. We assume that MN has been authenticated by CH_A with the help of the base station in a similar way to Han et al.'s schemes [9, 10] or Jiang et al.'s scheme [11]. After being authenticated by CH_A, MN shares information with CH_A, including a hashed value $H(I)$ of a random number I, a random number N_{MN}, and a pairwise key K_{CH_A-MN}. CH_A also generates AK_{MN} using its own KDK for reauthentication and passes it to MN:

$$AK_{MN} = f(KDK_{CH_A}, ID_{MN})$$

5.3 Phase 2: node reauthentication

Each cluster head CH_i periodically broadcasts beacon messages with current timestamp TS_1 to inform nodes of its presence:

$$CH_i \rightarrow * : ID_{CH_i}||TS_1$$

When MN moves to a new location and receives a beacon message from a foreign cluster head CH_B, it initiates the reauthentication procedure with CH_B, as shown in Fig. 3:

(1) MN computes v_1 and v_2, and sends the message "1" with current timestamp TS_2 to CH_B to rejoin the network:

$$MN \rightarrow CH_B : ID_{MN}||ID_{CH_A}||TS_2||v_1||v_2$$
$$v_1 = MAC(K_{MN-CH_A}, ID_{MN}||ID_{CH_A}||H(I))$$
$$v_2 = MAC(AK_{MN}, ID_{MN}||ID_{CH_A}||TS_2||TS_1||v_1)$$

(2) Upon receiving the message "1," CH_B generates AK_{MN} using KDK_{CH_A} and ID_{MN}:

$$AK_{MN} = f(KDK_{CH_A}, ID_{MN})$$

CH_B then verifies v_2 using AK_{MN} and authenticates MN. CH_B also checks whether or not TS_2 exceeds the specified time limit. If the result is valid, CH_B computes v_3 and sends the message "2" with current timestamp TS_3 to CH_A:

$$CH_B \rightarrow CH_A : ID_{MN}||TS_3||v_1||v_3$$
$$v_3 = MAC(K_{CH_A-CH_B}, ID_{MN}||TS_3||v_1)$$

Since AK_{MN} is only used to authenticate MN, the foreign cluster head CH_B must ask the home cluster head CH_A for the information required to generate a pairwise key K_{MN-CH_B} by sending the message "2."

(3) After receiving the message "2," CH_A verifies v_3 and v_1 using $K_{CH_A-CH_B}$ and K_{MN-CH_A}, respectively, and checks whether or not TS_3 exceeds the specified time limit. If the result is valid, CH_A computes e_1, which includes the information required to generate a pairwise key K_{MN-CH_B}, and v_4. CH_A then sends the message "3" with current timestamp TS_4 to CH_B:

$$CH_A \rightarrow CH_B : TS_4||e_1||v_4$$
$$e_1 = E(K_{CH_A-CH_B}, H(I)||N_{MN})$$
$$v_4 = MAC(K_{CH_A-CH_B}, TS_4||e_1)$$

(4) Upon receiving the message "3," CH_B verifies v_4 using $K_{CH_A-CH_B}$ and checks whether or not TS_4 exceeds the specified time limit. If the result is valid, CH_B obtains $H(I)$ and N_{MN} by decrypting e_1. CH_B then generates a random number N_{CH_B} and computes pairwise key K_{MN-CH_B}:

$$K_{MN-CH_B} = H(H(I)||N_{MN}||N_{CH_B})$$

For the next reauthentication, CH_B also generates a new AK AK'_{MN}:

$$AK'_{MN} = f(KDK_{CH_B}, ID_{MN})$$

CH_B then computes h_1, e_2, and v_5, and sends the message "4" with current timestamp TS_5 to MN:

$$CH_B \rightarrow MN : TS_5||h_1||e_2||v_5$$
$$h_1 = H(N_{MN}) \oplus N_{CH_B}$$
$$e_2 = E(K_{MN-CH_B}, AK'_{MN})$$
$$v_5 = MAC(K_{MN-CH_B}, TS_5||h_1||e_2)$$

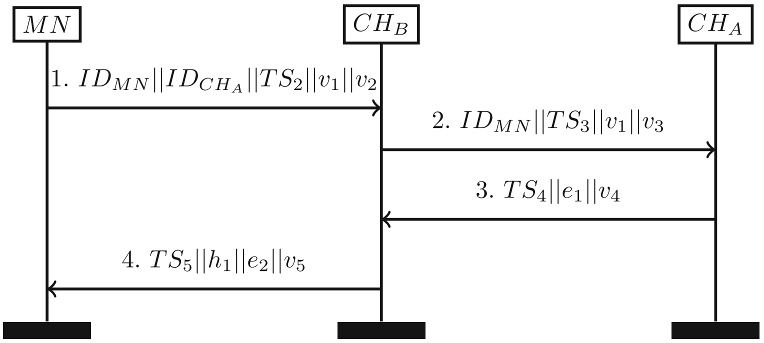

MN
CH_B
CH_A
1. $ID_{MN}||ID_{CH_A}||TS_2||v_1||v_2$
2. $ID_{MN}||TS_3||v_1||v_3$
3. $TS_4||e_1||v_4$
4. $TS_5||h_1||e_2||v_5$

Fig. 3 Overview of node reauthentication phase

(5) When MN receives the message “4” from CH_B, MN obtains N_{CH_B} from h_1 and computes a pairwise key K_{MN-CH_B} in the same way as CH_B. MN then verifies v_5 using this key and checks whether or not TS_5 exceeds the specified time limit. If the result is valid, MN obtains AK'_{MN} by decrypting e_2.

After completing the reauthentication phase, CH_B updates $H(I)$ and N_{MN} as $H(I')$ and N'_{MN}, respectively, for the next reauthentication, and sends them to MN during the communication process.

6 Security analysis

In this section, the security of ESMR is both formally and informally analyzed. First, we formally verified ESMR by modeling it using AVISPA tool [12]. Second, we conducted an informal security analysis of ESMR and confirmed that it meets security requirements and can prevent relevant security attacks.

6.1 Formal verification using AVISPA

AVISPA is a push-button tool that is widely used by many academic researchers to automatically validate various kinds of security protocols. The architecture of AVISPA is illustrated in Fig. 4.

In AVISPA, a security protocol is specified using a role-based formal language called a high-level protocol specification language (HLPSL). The HLPSL2IF translator translates the HLPSL specification into an intermediate format (IF). The IF specification is then validated by any of four back-end tools: OFMC, CL-AtSe, SATMC, and TA4SP under the Dolev-Yao intruder model [25]. Using these back-end tools, we can validate two kinds of security goals: secrecy and authentication. The secrecy goal is used to validate the confidentiality of information. In AVISPA, the secrecy goal is modeled using the goal predicate secret$(T, \text{id}, \{A, B\})$, which indicates that the value of term T is a secret shared only between agents A and B. The label id is used to identify the goal. The authentication goal is used to check whether or not two participants agree on a certain value in the current session. In AVISPA, the authentication goal is modeled using the goal predicates witness(B, A, id, T) and request(A, B, id, T) (for strong authentication) or wrequest(A, B, id, T) (for weak authentication). These predicates indicate that agent A authenticates agent B on some information T. The label id is used to identify the goal. The difference between strong and weak authentications is that weak authentication precludes replay attacks, but strong authentication does not. In this section, we briefly describe how we modeled ESMR in HLPSL and present the formal verification results of ESMR.

6.1.1 HLPSL specification of ESMR

Among the three phases of ESMR, the reauthentication phase, which is the main target phase of this study, was modeled and verified. We modeled a mobile node MN as *role_MN*, home cluster head CH_A as *role_CH_A*, and foreign cluster head CH_B as *role_CH_B*. Code block 1 presents the roles we modeled in the HLPSL. For the keyed message authentication code, we utilized a hash function by adding the symmetric key as one of the inputs. For example, v_1 is modeled as $MAC1' := MAC(M.H(Im).Kma)$, where Kma is the pairwise key K_{MN-CH_A} between MN and CH_A, and $MAC(\cdot)$ is a hash function called MAC.

Listing 1 HLPSL specification for roles

```
role role_MN(M,A,B:agent, H,MAC:hash_func,
    Kma:symmetric_key, AKm:message, Nm,Im:
    text, SND,RCV:channel(dy)) played_by M
    def=
 local
 State:nat, Tm,Tb1,Tb3:text, Nb:text,
```

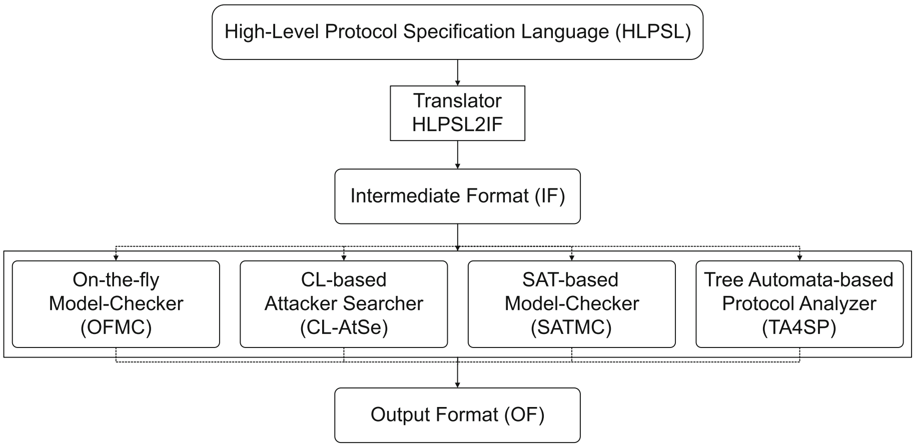

High-Level Protocol Specification Language (HLPSL)
Translator HLPSL2IF
Intermediate Format (IF)
On-the-fly Model-Checker (OFMC)
CL-based Attacker Searcher (CL-AtSe)
SAT-based Model-Checker (SATMC)
Tree Automata-based Protocol Analyzer (TA4SP)
Output Format (OF)

Fig. 4 AVISPA architecture

```
 Kmb,AKm2:message, MAC1,MAC2,MAC5:message

 init
 State := 0
 transition
 1. State=0/\RCV(B.Tb1') =|>
 State':=1/\Tm':=new()/\MAC1':=MAC(M.H(Im).
      Kma) /\MAC2':=MAC(M.A.Tm'.Tb1.MAC(M.H(
      Im).Kma).AKm)/\SND(M.A.Tm'.MAC1'.MAC2
      ')/\witness(M,B,b_m_v2,MAC2')/\witness
      (M,A,a_m_v1,MAC1')

 4. State=1
 /\RCV(Tb3'.xor(H(Nm),Nb').{AKm2'}_Kmb'.MAC
      (Tb3'.Tm.xor(H(Nm),Nb').{AKm2'}_Kmb'.H
      (Im).Kmb')) =|>
 State':=2/\Kmb':=H(H(Im).Nm.Nb')/\request(
      M,B,m_b_kmb,Kmb') end role

role role_CH_{A}(M,A,B:agent, H,MAC:
     hash_func, Kma,Kga,Kgb,Kab:
     symmetric_key, Nm,Im:text, SND,RCV:
     channel(dy)) played_by A def=
 local
 State:nat, Ta,Tb1:text, MAC1,MAC4:message

 init
 state := 0
transition
 2. State=0/\RCV(M.Tb2'.MAC(M.H(Im).Kma).
      MAC(M.Tb2'.MAC(M.H(Im).Kma).Kab)) =|>
 State':=1/\Ta':=new()/\MAC4':=MAC(Ta'.{H(
      Im).Nm}_Kab.Kab)/\SND(Ta'.{H(Im).Nm}
      _Kab.MAC4')/\MAC1':=MAC(M.H(Im).Kma)/\
      wrequest(A,M,a_m_v1,MAC1') end role

role role_CH_{B}(M,A,B:agent, H,MAC:
     hash_func, Kga,Kgb,Kab:symmetric_key,
    SND,RCV: channel(dy)) played_by B def=
 local
 State:nat, Tm,Ta,Tb1,Tb2,Tb3:text,
 Im,Nm,Nb:text, Kma:symmetric_key,
 Kmb,AKm2:message, MAC2,MAC3,MAC5:message

 const
 sec_kmb,sec_akm2:protocol_id

 init
 State := 0

 transition
 1. State=0/\RCV(start) =|>
 /\Tb1':=new()/\SND(B.Tb0')

 3. State=1/\RCV(M.A.Tm'.MAC(M.H(Im').Kma')
      .MAC(M.A.Tm'.Tb0.MAC(M.H(Im').Kma').H(
      Kga,M))) =|>
 State':=2/\Tb1':=new()/\MAC3':=MAC(M.Tb1'.
      MAC(M.H(Im').Kma').Kab)/\SND(M.Tb1'.
      MAC(M.H(Im').Kma').MAC3') /\ MAC2':=
      MAC(M.A.Tm'.Tb1'.MAC(M.H(Im').Kma').H(
      Kga,M))/\request(B,M,b_m_v2,MAC2')

 5. State=2/\RCV(Ta'.{H(Im).Nm'}_Kab.MAC(Ta
      '.{H(Im).Nm'}_Kab.Kab)) =|>
 State':=3/\Nb':=new()
 /\Kmb':=H(H(Im).Nm'.Nb')/\AKm2':=H(Kgb,M)
      /\Tb3':=new() /\ MAC5':=MAC(Tb3'.Tm.
      xor(H(Nm'),Nb').{AKm2'}_Kmb'.H(Im).Kmb
      ')
 /\SND(Tb3'.xor(H(Nm'),Nb').{AKm2'}_Kmb'.
      MAC5')/\witness(B,M,m_b_kmb,Kmb')/\
      secret(Kmb',sec_kmb,{M,B})/\secret(
      AKm2',sec_akm2,{M,A,B}) end role
```

In the HLPSL specification, three authentication and two secrecy goals are defined. For authentication goals, the mutual authentication between *MN* and CH_B, and the authentication of CH_A on *MN* are defined as follows:

(1) Upon receiving the message "1" from *MN*, CH_B authenticates *MN* through v_2. We use the label *b_m_v2* to identify the authentication goal. To verify the authentication goal, we add the witness and request predicates for the label *b_m_v2* to the roles of *MN* and CH_B, respectively.

(2) Upon receiving the message "2" from CH_B, CH_A authenticates *MN* through v_1. We use the label *a_m_v1* to identify the authentication goal. To verify the authentication goal, we add the witness and wrequest predicates for the label *a_m_v1* to the roles of *MN* and CH_A, respectively. Because CH_B checks the freshness of *MN*'s message and prevents the replay attack, CH_A only needs to perform weak authentication on *MN*.

(3) Upon receiving the message "5" from CH_B, *MN* authenticates CH_B using K_{MN-CH_B}. We use the label *m_b_kmb* to identify the goal. To verify the authentication goal, we add the witness and request predicates for the label *m_b_kmb* to the roles of CH_B and *MN*, respectively.

For secrecy goals, the secrecy of the pairwise key K_{MN-CH_B} and the secrecy of the authentication key AK'_{MN} are defined as follows:

(1) The pairwise key K_{MN-CH_B} should be only known to *MN* and CH_B. We use the label sec_*kmb* to identify the secrecy goal. To verify the secrecy of K_{MN-CH_B}, we add the secret predicate for the label sec_*kmb* to the role CH_B, where K_{MN-CH_B} is used.

(2) The authentication key AK'_{MN}, which is newly generated by CH_B, should only be known to *MN*, CH_A, and CH_B. We use the label sec_*akm*2 to identify the secrecy goal. To verify the secrecy goal, we add the secret predicate for the label sec_*akm*2 to the role *B*, where AK'_{MN} is generated.

Code block 2 presents the session and environment we modeled in the HLPSL. The environment section contains intruder knowledge and a composition of sessions. Because of the complexity of our model, we only defined two parallel sessions. Finally, we defined the goal facts to verify the three authentication goals and two secrecy goals outlined above.

Listing 2 HLPSL specification for session and environment

```
role session(M,A,B:agent, Kga,Kgb,Kma,Kab:
    symmetric_key, Im,Nm:text, H,MAC:
    hash_func) def=
 local
 AKm:message,
 SND1,RCV1,SND2,RCV2,SND3,RCV3:channel(dy)

 init
 AKm:=H(Kga,M)

 composition
 role_MN(M,A,B,H,MAC,Kma,AKm,Nm,Im,SND1,
     RCV1) /\
 role_CH_{A}(M,A,B,H,MAC,Kma,Kga,Kgb,Kab,Nm
     ,Im,SND2,RCV2) /\
 role_CH_{B}(M,A,B,H,MAC,Kga,Kgb,Kab,SND3,
     RCV3) end role

role environment() def=
 const
 node,cha,chb:agent, im,nm:text,
 kga,kgb,kma,kab:symmetric_key,
 hfunc,mac:hash_func,
 b_m_v2,a_m_v1,m_b_kmb:protocol_id

 intruder_knowledge = {node,cha,chb,hfunc,
     mac}

 composition
 session(node,cha,chb,kga,kgb,kma,kab,im,nm
     ,hfunc,mac) /\
 session(node,cha,chb,kga,kgb,kma,kab,im,nm
     ,hfunc,mac)

end role

goal
 authentication_on b_m_v2
 weak_authentication_on a_m_v1
 authentication_on m_b_kmb
 secrecy_of sec_kmb
 secrecy_of sec_akm2

end goal

environment()
```

6.1.2 *Formal verification results*

Figure 5 presents the formal verification results of our model. In ESMR, an exclusive-OR (XOR) operation is used. Among the four back-end tools, only OFMC and CL-AtSe support algebraic properties of operators, such as XOR operators and exponential operators. Therefore, two back-end tools, namely, OFMC and CL-AtSe, were used to verify our model. Figure 5a, b presents the formal verification results under OFMC and CL-AtSe, respectively. We confirmed that ESMR is safe under OFMC and CL-AtSe. Specifically, ESMR securely provides mutual authentication and pairwise key establishment between *MN* and CH_B, while preventing the replay attack. ESMR also prevents unconditional forwarding, which can used to launch DoS attacks on CH_A, because CH_B can authenticate *MN*.

6.2 Informal security analysis

We informally analyzed the security of ESMR in terms of satisfying security requirements and preventing relevant security attacks under the adversary model described in Section 4.2. Table 2 compares the security of ESMR with schemes proposed by Han et al. [9, 10] and Jiang et al. [11].

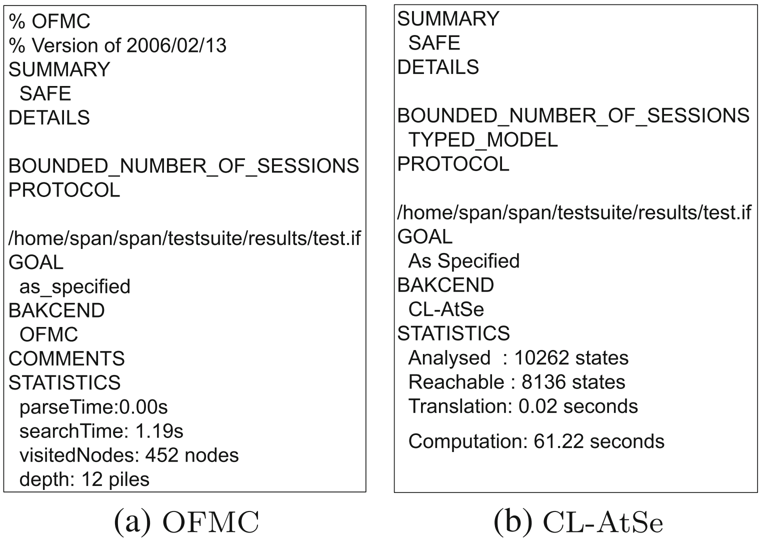
Mutual authentication ESMR supports mutual authentication between mobile nodes and cluster heads. In ESMR, CH_B authenticates mobile node *MN* by verifying v_2 using AK_{MN}. Since CH_B is one of the neighbors of CH_A, it has KDK_{CH_A} and can compute AK_{MN}. When CH_A receives the message "2" from CH_B, it also authenticates *MN* by verifying v_1 using K_{MN-CH_A} and $H(I)$. If one of neighbors of CH_A is captured, KDK_A is exposed to the adversary. The adversary can then utilize the exposed KDK_A to make an illegal node to bypass the authentication of CH_B. However, because CH_A authenticates the illegal node again using K_{MN-CH_A} and $H(I)$, the illegal node cannot join the network even if it bypasses the authentication of CH_B. *MN* authenticates CH_B using N_{MN} and $H(I)$. When MN receives the message "4" from CH_B, it first obtains N_{CH_B} from h_1 and computes K_{MN-CH_B} using $H(I)$, N_{MN}, and N_{CH_B}. *MN* then verifies v_5 using K_{MN-CH_B} and $H(I)$. Thus, ESMR satisfies the mutual authentication.

Key freshness In ESMR, a new pairwise key is generated whenever *MN* moves and is connected to a new cluster head. *MN* and CH_B generate a pairwise key K_{MN-CH_B} using $H(I)$, N_{MN}, and N_{CH_B}. Since $H(I)$ and N_{MN} are updated after reauthentication is completed and N_{CH_B} is freshly generated for each session, a new pairwise key is generated for each session. Thus, ESMR satisfies key freshness.

Replay attack prevention In ESMR, all messages contain timestamps to prevent replay attacks. Thus, the network is assumed to be loosely synchronized. If the timestamp of a received message exceeds a specified time limit, it is determined to be a potential replay attack and the message is dropped. Since a new pairwise key is generated each session, this effectively prevents the replay attack.

Outsider DoS attack prevention DoS attacks can be launched by inside adversary and outside adversary. The inside adversary can launch DoS attacks by capturing the mobile node or cluster head, and replicating them. Since the replicated nodes have cryptographic materials such as secret keys, general security mechanisms such as authentication and encryption cannot prevent and detect insider DoS attacks. Thus, all the four schemes are vulnerable against insider DoS attacks.

Authentication is the first step to detect and prevent DoS attacks from outside adversary. However, in Jiang et



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS

BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL

/home/span/span/testsuite/results/test.if
GOAL
  as_specified
BAKCEND
  OFMC
COMMENTS
STATISTICS
  parseTime:0.00s
  searchTime: 1.19s
  visitedNodes: 452 nodes
  depth: 12 piles
```

(a) OFMC

```
SUMMARY
  SAFE
DETAILS

BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL

/home/span/span/testsuite/results/test.if
GOAL
  As Specified
BAKCEND
  CL-AtSe
STATISTICS
  Analysed  : 10262 states
  Reachable : 8136 states
  Translation: 0.02 seconds

  Computation: 61.22 seconds
```

(b) CL-AtSe

Fig. 5 Formal verification results. **a** OFMC **b** CL-AtSe

al.'s scheme [11], since there are no shared secrets between foreign cluster heads and mobile nodes, a foreign cluster head cannot verify the reauthentication request of a mobile node and unconditionally forwards it to the home cluster head. This leads to DoS attacks on the home cluster head. In contrast, ESMR can prevent unconditional forwarding, thus preventing potential DoS attacks based on unconditional forwarding. In ESMR, the message "1," sent by MN to CH_B, contains the message authentication code v_2 generated using AK_{MN}. Because CH_B is one of neighbors of CH_A, it can compute AK_{MN} using KDK_{CH_A} and ID_{MN}. CH_B then verifies v_2 using AK_{MN} and authenticates MN. Thus, ESMR prevent DoS attacks based on unconditional forwarding by allowing the foreign cluster head to authenticate the mobile node directly.

Compromise resilience It refers that even if a node is captured by an adversary, the compromised node reveals no information about links it is not directly involved with. Compromise resilience is high if an adversary cannot deduce the cryptographic secrets of any other nodes not directly involved with the compromised node. However, if even a single-cluster head is captured in Han et al.'s schemes [9, 10], multiple TGKs are exposed to the adversary. The adversary can then obtain the secret information included in all tickets generated using the exposed TGKs and compromise the communication security for multiple nodes. Although ESMR uses the KDK as a group key in a similar way to the TGK in Han et al.'s schemes, the KDK is only used to generate AK, which are only used by CH_B to authenticate mobile nodes. Therefore, even if a cluster head is captured and multiple KDKs are exposed to the adversary, an adversary cannot obtain any pairwise keys other than those between the compromised cluster head and mobile node. Thus, ESMR provides higher-compromise resilience than Han et al.'s schemes.

Table 2 Security comparison between ESMR and related schemes

	[9]	[10]	[11]	ESMR
Mutual authentication	Yes	Yes	Yes	Yes
Key freshness	Yes	Yes	Yes	Yes
Replay attack prevention	Yes	Yes	Yes	Yes
Outsider DoS attack prevention	Yes	Yes	No	Yes
Compromise resilience	Low	Low	High	High
Forward security	No	No	Yes	Yes

Forward security It refers that even if a node is captured and its current secrets are leaked, an adversary cannot decrypt any data collected and encrypted before the compromise. In ESMR, the pairwise key between MN and CH_B is generated using freshly generated random numbers for each reauthentication; hence, pairwise keys are independent of each other. In other words, even if an adversary obtains the current pairwise key by compromising MN, the adversary cannot derive the previous pairwise key. When a cluster head is captured, multiple KDKs are exposed to the adversary. However, the KDK is only used to generate AKs used for mobile node reauthentication. Therefore, even if an adversary obtains the KDK by compromising the cluster head, the adversary cannot derive any pairwise keys. Thus, ESMR satisfies forward security.

In contrast, Han et al.'s schemes [9, 10] do not satisfy the forward security. They employ ticket for reauthentication, containing the previous pairwise key encrypted with the TGK. Therefore, if an adversary obtains the TGK by capturing the cluster head, the adversary can obtain the previous pairwise key from the ticket and decrypt any previous messages encrypted using that key.

7 Performance evaluation

We evaluated the performance of ESMR by comparing it with schemes proposed by Han et al. [9] and Jiang et al. [11] in terms of energy consumption and reauthentication latency. We also analyzed the performance of ESMR under DoS attacks based on unconditional forwarding by comparing it with Jiang et al.'s scheme [11] in terms of reauthentication latency and packet delivery ratio.

7.1 Evaluation methodology

The performance of the three schemes have been evaluated by means of simulation experiments. Simulations were performed by using the OMNeT++ simulator with INET framework version 3.6.3 to measure energy consumption and reauthentication latency. For the simulation, we considered the situation where a mobile node has moved to a new location and initiates the reauthentication procedures with a foreign cluster head.

The simulation sets up IEEE 802.15.4 using the IEEE 802.15.4 narrow band network interface card module provided by the INET framework and Tmote sky datasheet [26]. The data transmission speed was set to 250 kbps and the receiver sensitivity was set to − 95 dBm. The transmission power of the mobile node and cluster head were set to − 10 dBm and 0 dBm, respectively. The mobile node and cluster head had communication ranges of 30 m and 100 m, respectively, measured by the INET framework. The simulation used static routing, the simplest form of routing. Message size was calculated based on the following base parameter settings: ID of 2 bytes, MAC of 4 bytes, timestamp of 8 bytes, random number of 8 bytes, and key size of 16 bytes. It is also assumed that encryption does not change the message size.

7.2 Energy consumption analysis result and discussion

Because all the three schemes use symmetric key cryptography, there is no significant difference in energy consumption due to computations. Therefore, only energy consumption arising from communication was measured and compared. Since communication is generally the major energy consumption, this comparison is sufficient to investigate energy efficiencies of the three schemes. Based on the Tmote sky datasheet [26], the receiving current consumptions for the mobile node and cluster head were all set to 19.7 mA and the transmitting current consumptions for them were set to 11.2 and 17.4 mA, respectively. The simulation was conducted for five scenarios where the number of hops (n) between the mobile node and cluster head varies from one to five.

Figure 6 presents comparisons of the total energy consumption for a single reauthentication based on the number of hops (n) between the mobile node and cluster head. The total energy consumption is calculated as the sum of the energy consumed by all nodes participating in the reauthentication process.

When $n = 1$, ESMR has the greatest total energy consumption among the three schemes. ESMR has the same reauthentication process as Jiang et al.'s scheme, but has a larger total message size. Consequently, ESMR incurs 3% more total energy consumption than Jiang et al.'s scheme. In Han et al.'s scheme, the mobile node and cluster head authenticate each other and establish a pairwise key using a ticket without the help of other nodes. In contrast, in ESMR, the information required for reauthentication is exchanged between the foreign cluster head and home cluster head. Consequently, ESMR incurs 16% more total energy consumption than Han et al.'s scheme.

However, when $n \geq 2$, Han et al.'s scheme has the greatest total energy consumption among the three schemes. The message size that the mobile node sends to the cluster head is larger than for Jiang et al.'s scheme and ESMR by at least 30 bytes. Hence, the energy consumption of Han et al.'s stationary node, which relays the mobile node's message to the cluster head, is larger than for Jiang et al.'s scheme and ESMR. Consequently, Han et al.'s scheme has greater total energy consumption than Jiang et al.'s scheme and ESMR. ESMR also incurs up to 6% more total energy consumption than Jiang et al.'s scheme when $n \geq 2$, because ESMR has the same reauthentication process as Jiang et al.'s scheme, but a larger total message size.

Thus, ESMR is suitable for multi-hop communication environment in terms of energy consumption, where the number of hops between the mobile node and cluster

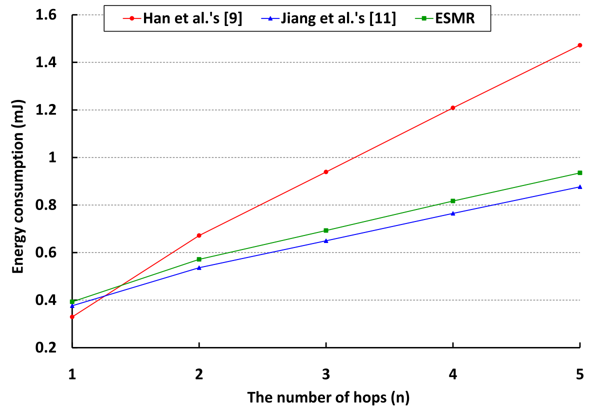


Fig. 6 Total energy consumption for a single reauthentication

head is two or more. Specifically, ESMR has only up to 6% increase in total energy consumption compared with existing schemes, which is negligible.

7.3 Reauthentication latency analysis result and discussion

Since the CC2420 featured by Tmote sky provides hardware support for AES-128, the computational delays caused by encryption and decryption are very small and were not considered. For example, the time required to encrypt 16 bytes is 449.203 μs for Tmote sky using CC2420 hardware encryption [27]. Therefore, only reauthentication latency as a result of communication was measured. Five scenarios were considered for the simulation where the number of hops (n) between the mobile node and cluster head varies from one to five. Simulations were conducted 100 times per scenario for each scheme.

Figure 7 compares reauthentication latency when $n = 1$. Average reauthentication latencies of Han et al.'s scheme, Jiang et al.'s scheme, and ESMR were 10.4 ms, 11.8 ms, and 12.2 ms, respectively. ESMR has the longest average reauthentication latency since it requires the same number of messages as Jiang et al.'s scheme, but has larger total message size. Thus, ESMR has 3% longer average reauthentication latency than Jiang et al.'s scheme. ESMR also requires communication between cluster heads to exchange the information required for reauthentication. In contrast, Han et al.'s scheme does not require communication between cluster heads because the mobile node and cluster head authenticate each other and establish pairwise keys using the ticket. Consequently, ESMR has 16% longer average reauthentication latency than Han et al.'s scheme. However, the actual difference in average reauthentication latency between ESMR and Han et al.'s scheme is very small at 1.8 ms. Moreover, since average reauthentication latency for all the three schemes is less than 13 ms, the three schemes are sufficiently fast.

Figure 8 compares reauthentication latency when $n = 5$. Average reauthentication latencies of Han et al.'s scheme, Jiang et al.'s scheme, and ESMR were 39.6 ms, 25.4 ms, and 26.2 ms, respectively. Han et al.'s scheme has the longest reauthentication latency since the message size that mobile nodes send to cluster heads is larger than for Jiang et al.'s scheme and ESMR. Han et al.'s scheme also requires an additional message for reauthentication compared with Jiang et al.'s scheme and ESMR; hence, reauthentication latency of a stationary node that relays a mobile node's message to a cluster head is longer than for Jiang et al.'s scheme and ESMR. ESMR has 3% longer average reauthentication latency than Jiang et al.'s scheme because although ESMR requires the same number of messages for reauthentication, it has a larger total message size than Jiang et al.'s scheme.

Figure 9 compares average reauthentication latency based on the number of hops (n) between the mobile node and cluster head. Han et al.'s scheme has the shortest reauthentication latency when $n = 1$, but the longest when $n \geq 2$. ESMR has slightly longer reauthentication latency than Jiang et al.'s scheme regardless of n due to the larger message size. However, the actual difference is very small which is less than 0.9 ms. Thus, Jiang et al.'s scheme and ESMR have similar reauthentication latency regardless of n.

Thus, in terms of reauthentication latency, ESMR is suitable for both single-hop and multi-hop communication environments. Specifically, when $n = 1$, ESMR has a maximum increase in the average reauthentication latency of 1.8 ms compared with existing schemes, but it is fast enough because the actual average reauthentication latency is less than 13 ms, like the existing schemes.

7.4 Performance analysis under DoS attacks based on unconditional forwarding

We considered a scenario in which outside adversary sends a huge amount of spoofed or altered mobile nodes' reauthentication requests to neighboring cluster heads of a home cluster head in order to attempt DoS attacks based on unconditional forwarding on the home cluster head. We also assumed that each adversary node sends the messages only to one of neighboring cluster heads of the home

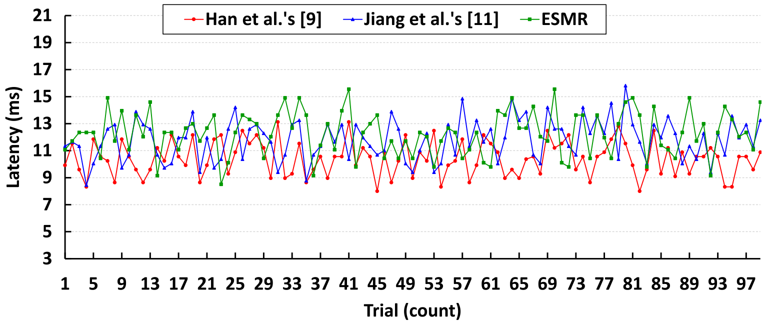


Fig. 7 Reauthentication latency when number of hops $n = 1$

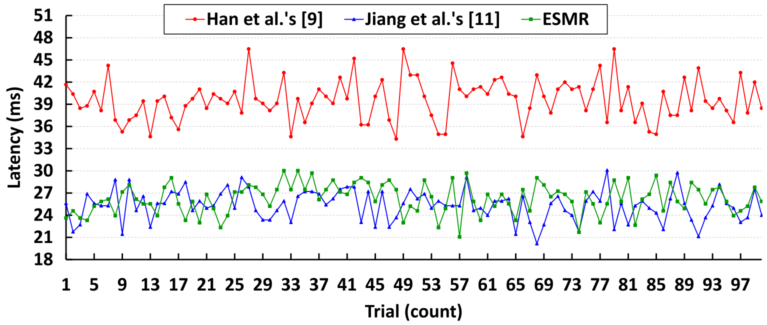


Fig. 8 Reauthentication latency when $n = 5$

cluster head except for the foreign cluster head. Hop distance between the mobile node and home cluster head was set to 1, and message transmission interval of adversary nodes was set to 10 ms. Three cases were considered for the simulation where the number of adversary nodes (k) varies from one to three. Simulations were conducted 500 times per case for Jiang et al.'s scheme [11] and ESMR.

Figure 10 compares packet delivery ratio under DoS attacks based on unconditional forwarding according to the number of adversary nodes (k). The packet delivery ratio of ESMR was 100% regardless of k. On the other hand, the packet delivery ratio of Jiang et al.'s scheme decreases from 100 to 11% as k increases from 0 to 3.

Figure 11 compares average reauthentication latency under DoS attacks based on unconditional forwarding according to the number of adversary nodes (k). The average reauthentication latency of ESMR was approximately 12 ms regardless of k. On the other hand, the average reauthentication latency of Jiang et al.'s scheme increases from 11.8 ms to 24 ms as k increases from 0 to 3. Although the packet delivery ratio of Jiang et al.'s scheme was 100% when $k = 1$, the average reauthentication latency was 20.2 ms which is 28.5% longer than for ESMR.

Thus, ESMR can effectively prevent DoS attacks based on unconditional forwarding on the home cluster head because it can prevent unconditional forwarding itself by allowing the neighboring cluster heads to verify validity of the spoofed or altered reauthentication requests, i.e., the neighboring cluster heads do not forward invalid reauthentication requests to the home cluster head unlike Jiang et al.'s scheme.

8 Conclusion

Several mobile node reauthentication schemes based on symmetric key cryptography have been proposed to efficiently handle frequent reauthentication [9–11]. However, we found that Han et al.'s schemes [9, 10] do not provide high-compromise resilience and Jiang et al.'s scheme [11] have a problem of unconditional forwarding which can be used to launch DoS attack on the home cluster head.

In this paper, we proposed the energy-efficient and secure mobile node reauthentication (ESMR) for MWSNs, which satisfies the security requirements of MWSNs by addressing the security weaknesses of the existing schemes [9–11]. Security analysis verified that ESMR meets the security requirements of MWSNs and

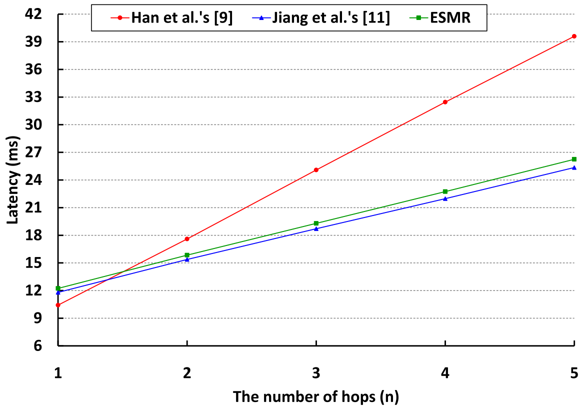


Fig. 9 Average reauthentication latency

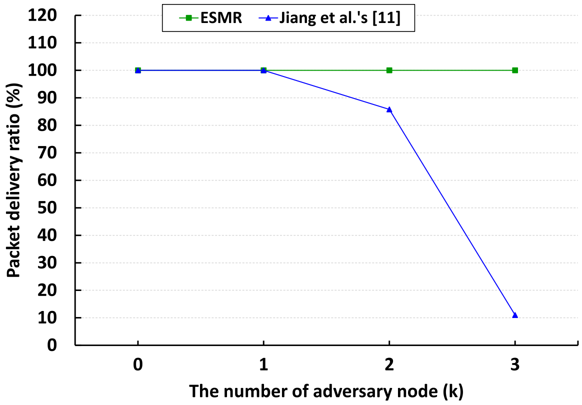


Fig. 10 Packet delivery ratio under DoS attacks based on unconditional forwarding

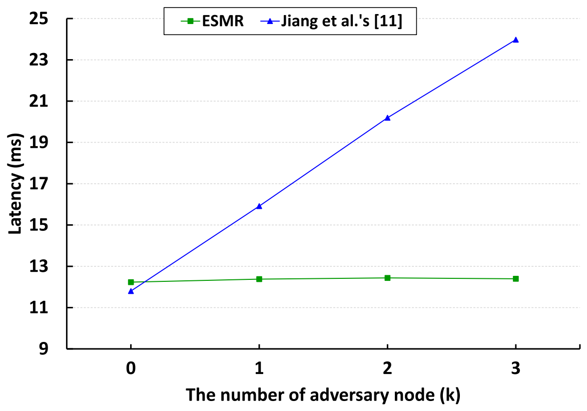


Fig. 11 Average reauthentication latency under DoS attacks based on unconditional forwarding

can prevent relevant security attacks. ESMR is suitable for multi-hop communication environment, but ESMR can be applied to single-hop communication environment. ESMR requires less than 16% increased total energy consumption when the number of hops between the mobile node and cluster head is one, but ESMR is fast enough because the average reauthentication latency is less than 13 ms, like the existing schemes. Moreover, ESMR meets the security requirements of MWSNs by addressing the security weaknesses of the existing scheme. Thus, ESMR can provide secure and fast mobile node reauthentication with slightly increased total energy consumption for single-hop communication environment.

Abbreviations

AES: Advanced encryption standard; AK: Authentication key; AVISPA: Automated validation of internet security protocols and applications; CL-AtSe: CL-based attacker searcher; CL-HSC: Certificateless hybrid signcryption scheme; DoS: Denial of service; ECC: Elliptic curve cryptography; HLPSL: High-level protocol specification language; IF: Intermediate form; IoT: Internet of things; KDK: Key derivation key; MWSNs: Mobile wireless sensor networks; NCL: Neighboring cluster head list; OFMC: On-the-fly model-checker; OF: Output format; SATMC: SAT-based model-checker; TA4SP: Tree automata-based protocol analyzer; TGK: Ticket generation key; WSNs: Wireless sensor networks; XOR: Exclusive-OR

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2019R1I1A1A01062743).

Funding

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2019R1I1A1A01062743).

Availability of data and materials

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Authors' contributions

BK proposed the main concept, conducted formal verification and simulations, analyzed results, and wrote the manuscript. JS contributed to revising the manuscript and fine-tuning the proposed scheme. Both authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 10 June 2018 Accepted: 7 May 2019
Published online: 11 June 2019

References

1. X. Wang, S. Han, Y. Wu, X. Wang, Coverage and energy consumption control in mobile heterogeneous wireless sensor networks. IEEE Trans. Autom. Control. **58**(4), 975–988 (2013)
2. Y. Yang, M. I. Fonoage, M. Cardei, Improving network lifetime with mobile wireless sensor networks. Comput. Commun. **33**(4), 409–419 (2010)
3. O. Chipara, C. Lu, T. C. Bailey, G.-C. Roman, in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems. SenSys '10*. Reliable clinical monitoring using wireless sensor networks: Experiences in a step-down hospital unit (ACM, New York, 2010), pp. 155–168
4. S. Ehsan, K. Bradford, M. Brugger, B. Hamdaoui, Y. Kovchegov, D. Johnson, M. Louhaichi, Design and analysis of delay-tolerant sensor networks for monitoring and tracking free-roaming animals. IEEE Trans. Wirel. Commun. **11**(3), 1220–1227 (2012)
5. M. Li, Y. Liu, Underground coal mine monitoring with wireless sensor networks. ACM Trans. Sens. Netw. (TOSN). **5**(2), 10 (2009)
6. C. Zhu, L. Shu, T. Hara, L. Wang, S. Nishio, L. T. Yang, A survey on communication and data management issues in mobile sensor networks. Wirel. Commun. Mob. Com. **14**(1), 19–36 (2014)
7. A. Ghosal, S. Halder, in *Cooperative Robots and Sensor Networks 2015*. Security in mobile wireless sensor networks: Attacks and defenses (Springer International Publishing, Cham, 2015), pp. 185–205
8. A. Achour, L. Deru, J. C. Deprez, Mobility management for wireless sensor networks a state-of-the-art. Procedia Comput. Sci. **52**, 1101–1107 (2015)
9. K. Han, K. Kim, T. Shon, Untraceable mobile node authentication in wsn. Sensors. **10**(5), 4410–4429 (2010)
10. K. Han, T. Shon, K. Kim, Efficient mobile sensor authentication in smart home and wpan. IEEE Trans. Consum. Electron. **56**(2), 591–596 (2010)
11. S. Jiang, J. Zhang, J. Miao, C. Zhou, A privacy-preserving reauthentication scheme for mobile wireless sensor networks. Int. J. Distrib. Sens. Netw. **9**(5), 913782 (2013)
12. A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P. C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, L. Vigneron, in *Computer Aided Verification*. The avispa tool for the automated validation of internet security protocols and applications (Springer, Berlin, Heidelberg, 2005), pp. 281–285
13. M. Bilal, S.-G. Kang, An authentication protocol for future sensor networks. Sensors. **17**(5), 979 (2017)
14. Y. Qiu, J. Zhou, J. Baek, J. Lopez, Authentication and key establishment in dynamic wireless sensor networks. Sensors. **10**(4), 3718–3731 (2010)
15. S. H. Erfani, H. H. S. Javadi, A. M. Rahmani, A dynamic key management scheme for dynamic wireless sensor networks. Secur. Commun. Netw. **8**(6), 1040–1049 (2015)
16. F. Gandino, C. Celozzi, M. Rebaudengo, A key management scheme for mobile wireless sensor networks. Appl. Sci. **7**(5), 490 (2017)
17. X. Zhang, J. He, Q. Wei, Eddk: Energy-efficient distributed deterministic key management for wireless sensor networks. EURASIP J. Wirel. Commun. Netw. **2011**, 1–11 (2011)
18. S.-H. Seo, J. Won, S. Sultana, E. Bertino, Effective key management in dynamic wireless sensor networks. IEEE Trans. Inf. Forensic. Secur. **10**(2), 371–383 (2015)
19. M. Omar, I. Belalouache, S. Amrane, B. Abbache, Efficient and energy-aware key management framework for dynamic sensor networks. Comput. Electr. Eng. **72**, 990–1005 (2018)
20. H. Chan, A. Perrig, D. Song, in *2003 Symposium on Security and Privacy*. Random key predistribution schemes for sensor networks (IEEE, Berkeley, 2003), pp. 197–213

21. L. Eschenauer, V. D. Gligor, in *Proceedings of the 9th ACM Conference on Computer and Communications Security. CCS '02*. A key-management scheme for distributed sensor networks (ACM, New York, 2002), pp. 41–47
22. A. Gonga, O. Landsiedel, M. Johansson, in *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*. MobiSense: Power-efficient micro-mobility in wireless sensor networks (IEEE, Barcelona, 2011), pp. 1–8
23. M. Nabi, M. Blagojevic, M. Geilen, T. Basten, T. Hendriks, in *2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. Mcmac: An optimized medium access control protocol for mobile clusters in wireless sensor networks (IEEE, Boston, 2010), pp. 1–9
24. Q. Dong, W. Dargie, A survey on mobility and mobility-aware mac protocols in wireless sensor networks. IEEE Commun. Surv. Tutor. **15**(1), 88–100 (2013)
25. D. Dolev, A. Yao, On the security of public key protocols. IEEE Trans. Inf. Theory. **29**(2), 198–208 (1983)
26. Tmote Sky Datasheet (2006). http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf. Accessed 23 Jan 2018
27. M. Healy, T. Newe, E. Lewis, in *Smart Sensors and Sensing Technology*. Analysis of hardware encryption versus software encryption on wireless sensor network motes (Springer, Berlin, Heidelberg, 2008), pp. 3–14