Lightweight Key Renewals for Clustered Sensor Networks

Gicheol Wang
KISTI/Dept. of Computing and Networking Resources, Daejeon, Korea
Email: gcwang@kisti.re.kr

Seunghae Kim
KISTI/Dept. of Computing and Networking Resources, Daejeon, Korea
Email: shkim@kisti.re.kr

Daewook Kang, Deokjai Choi
Chonnam Nat’l University/School of Electronic and Computer Engineering, Gwangju, Korea
Email: {dwkang, dchoi}@chonnam.ac.kr

Gihwan Cho
Chonbuk Nat’l University/CAIIT, Div. of Electronics and Information Engineering, Jeonju, Korea
Email: ghcho@chonbuk.ac.kr

Abstract— In sensor networks, sensors are likely to be captured by attackers because they are usually deployed in an unprotected or even a hostile environment. If an adversarial compromises a sensor, he/she uses the keys from the compromised sensor to uncover the keys of others sensors. Therefore, it is very important to renew the keys of sensors in a proactive or reactive manner. Even though many group key renewal schemes have been proposed, they have some security flaws. First, they employ a single group key in a cluster so that the compromise of one sensor discloses the group key. Second, they evict the compromised nodes by updating the compromised keys with non-compromised keys. This eviction scheme is useless when the non-compromised keys are exhausted due to the increase of compromised nodes. In this paper, we propose a lightweight key renewal scheme, which evicts the compromised nodes clearly by reforming clusters excluding compromised nodes. Besides, in a cluster, each member employs a pairwise key for communication with its CH (Cluster Head) so that our scheme is tolerable against sensor compromise. Our simulation results prove that the proposed scheme is more energy-saving than the group key renewal schemes.

Index Terms—key management, cluster organization, wireless sensor networks, confidentiality, integrity, energy efficiency

I. INTRODUCTION

Wireless sensor networks are expected to realize various applications such as pollution detection, battlefield reconnaissance, environmental protection, and disaster relief [1].

Sensor nodes are usually deployed in an unprotected or even an adversarial environment [2]. It makes the sensor network need a protection mechanism of transmissions from sensors to the sink which is also called as BS (Base Station). For the protection of sensor readings, using cryptographic keys between sensors is the cheapest and the most feasible solution. However, the key management in sensor networks is not a trivial problem because the sensors have many constraints such as limited battery lifetime and small-space memory [1-4].

Recently, cluster structure has been frequently used in sensor networks to distribute the key management duty to some designated nodes. The designated nodes are called as Cluster Heads (CHs) and each normal sensor is served by only one CH. More importantly, a CH plays the role of local key management server so that the cluster structure enables the distributed key management service.

In sensor networks, sensors are likely to be captured by attackers, and all keys inside the captured sensor are revealed to the attackers. We call these sensors as compromised sensors in this paper. Besides, they can be usefully employed for revealing the keys inside other sensors. To mitigate this problem, the exposed keys should be renewed hiding the renewal process from the captured sensors. This remedy reduces the activity time of attackers, and it is called as key renewal scheme.

Many key renewal schemes have been proposed so far. They can be classified into two schemes; no renewal schemes [5-9] and group key renewal schemes [10-14]. In no renewal schemes, some administrative keys are pre-
assigned from a central key management server (that is, the BS), and the keys used for communications are generated using the pre-assigned keys. The pre-assigned keys are never renewed until the extinction of network. These pre-assigned keys are taken by other sensors with a predefined probability, so these schemes severely degrade the security of the network as the number of compromised sensors increases. Fig. 1(a) shows the procedures of the no renewal schemes. In group key renewal schemes, the administrative keys are assigned from a local key management server (that is, CH) after deployment. A CH generates a group key for communications in the cluster and distributes it to members using the administrative keys. The post-assigned keys and the group key are periodically renewed to prevent cryptanalysis attacks, and they are also renewed when a CH detects a compromised sensor in its cluster. In the group key renewal schemes, the number of administrative keys employed in a cluster is much smaller than that of no renewal schemes. As a result, if the number of compromised sensors increases, the key renewal method using the post-assigned keys has no effect. Fig. 1(b) shows the procedure of the group key renewal scheme.

In this paper, we propose a lightweight renewal scheme which realizes the key renewals by dynamic cluster organization. In our scheme, each sensor establishes pairwise keys with its neighbors using its ID and the network-wide key at network boot-up time. Some of the pairwise keys are selected for intra-cluster communication, and the cluster organization makes the selection of the pairwise keys. Our scheme periodically refreshes the cluster organization instead of renewing the pairwise keys. As a result, new pairwise keys are selected for intra-cluster communication, and newly elected CHs agree a CH-BS key by informing the BS of their membership information.

This paper is organized as follows. Section II deals with previous related work. Section III describes our scheme in detail. We analyze the security of our scheme in Section IV and provide the simulation results in Section V. In Section VI, we examine how the extension of neighbor radius affects the performance of our scheme. We deal with the way of detecting the compromised sensors in Section VII. We conclude this paper in Section VIII.

II. RELATED WORK

Eschnauer and Gligor firstly proposed a key management scheme which is based on the key pre-distribution [9]. In this scheme, each sensor is assigned a predefined number of keys before the network deployment. After the deployment, each sensor establishes communication keys with neighbors using the common assigned keys. If there are no common keys between any two sensors, they can indirectly establish communication keys using neighbors sharing common keys. However, the scheme is vulnerable to sensor compromise, since it allows the establishment of communication keys even when the number of common keys is one. To resolve this problem, Chan et al. allowed the communication key establishment only if the number of common keys is more than q (>1) keys [5]. Du et al. showed that the pre-distributed keys can be shrunked, if approximate locations in the deployment area are known to the key pre-distribution server [6]. The key pre-distribution server partitions the deployment area into a specific number of sub areas and distributes some keys to the key pool of each sub area. The key pre-distribution procedure is adjusted, thereby nearby sub areas share a
lot of common keys. Therefore, although the pre-distributed keys at a sensor shrinks, any two nearby sensors can easily find common keys. Liu et al. proposed a key pre-distribution scheme in which sensors are deployed in groups [7]. Because sensors belonging to the same group are close to each other, the key pre-distribution server distributes some keys to each sensor so that the rate of sharing common keys in the same group is very high. If some nearby sensors belong to different groups, such sensors act as an inter-group gateway to support the establishment of path keys between sensors belonging to multiple groups. Traynor et al. proposed a key pre-distribution scheme in which high and low capability sensors are mixed [8]. High capability sensors hold much more keys than low capability sensors. Also, high capability sensors support communication of low capability sensors, thereby enabling a hierarchical communication model. Because low capability sensors hold a small number of keys, this scheme obtains the saving of memory space and the improvement of resiliency against sensor compromise.

Above schemes do not have any key renewal mechanisms in the key management process. Therefore, key materials obtained from captured sensors are valid until the extinction of the network, and the compromised key materials also exist inside many other sensors with a predefined probability. As a result, the security of above schemes is threatened by attackers if the number of captured sensors increases.

Eltoweissy et al. proposed an EBS (Exclusion Basis System) which defends a group key against evicted nodes in a communication group [12]. An EBS system holds \( k + m \) administrative keys to secure the group key update, and each group member obtains \( k \) administrative keys from the EBS system. If a group member is evicted, updating the current group key needs only \( m \) messages. Because the evicted nodes do not have \( m \) keys encrypting \( m \) messages, they cannot participate in communications in the group any longer. In [13], Eltoweissy et al. applied the EBS system to the group key management in wireless sensor networks. First, each sensor determines its cluster through the training content broadcasted by the BS. Each cluster manages a group key, and EBS system is applied to the network to update the group keys. Jolly et al. proposed a dynamic key management scheme which is not based on the EBS system [11]. The BS is responsible for key generation and assignment while gateways (CHs) are responsible for obtaining some needed keys through communication between neighboring gateways. This scheme reduces memory overhead at sensors (generally, two keys), but increases communication overhead significantly. This is because rekeying in this scheme causes cluster reorganization and key redistribution. Younis et al. pointed out that Eltoweissy’s EBS system is vulnerable to collusion attacks launched by a group of attackers [14]. In order to resolve the problem, Younis et al. proposed a scheme, which was called SHELL (Scalable, Hierarchical, Efficient, Location-aware, and Lightweight). SHELL performs location-based key assignment in a cluster to decrease the number of keys revealed by the collusion of attackers. That is, nearby sensors in SHELL share more common administrative keys than distant sensors. SHELL performs EBS-based rekeying, and the rekeying occurs only within each cluster. Eltoweissy proposed a scheme, so called LOCK (LOcated Combinatorial Keying), where EBS system is employed for rekeying not only between CH and sensors but also between BS and CHs [10]. Panja et al. proposed a dynamic key management scheme in which all sensors in a cluster contribute to the generation and renewal of the group key [3]. In a cluster, all member sensors generate partial secret keys and send them to CH, and the CH generates a group key using the partial secret keys. Then, the CH distributes the group key to all members using a pre-shared key. If a renewal of the group key is required, the CH first renews the group key by adding or removing a specific partial key(s), and informs members of the addition and removal of the specific partial key(s). Landstra et al. improved the energy-efficiency of Panja’s scheme [1]. In the scheme, the number of partial secret keys needed for secure group key generation is known to all sensors. Thanks to the number, a CH generates a group key using partial keys of some sensors, not all sensors. First, the CH determines the sub clusters which take part in the generation of a group key. The number of sub clusters depends on the number of partial secret keys needed for group key generation. Then, only the sensors in the sub clusters participate in a group key generation. Therefore, it can reduce the delay time and energy which are consumed for the group key generation and renewal.

Most of above schemes have the group key renewal mechanisms in the key management process. A key administrative server detects the compromise of sensors itself or via the report of a veracious sensor(s) and performs the renewal of the compromised keys using the non-compromised keys. Because this key renewal process mainly relies on the non-compromised keys, the decrease of the non-compromised keys (that is, the increase of the compromised sensors) threatens the security of the renewal mechanism. That is, if the number of compromised sensors increases, neighboring compromised sensors can reveal the renewal process by colluding with each other. As a result, some compromised sensors can uncover the renewed group key and keep doing the malicious actions such as eavesdropping and fabricating data from other sensors.

III. LIGHTWEIGHT KEY RENEWAL SCHEME

A. System Model and Assumptions

We employ the following system model.

- Once sensors are excluded from network due to energy exhaustion, they are never reused. Instead, some additional sensors are deployed in the duty field to substitute the excluded sensors.
- The network consists of a BS and a large number of quasi-stationary sensors, and it is partitioned into clusters on the basis of a rule.
- The BS assigns a single shared key (which is called network-wide key in this paper) and some
administrative keys to each sensor before deployment.
- All sensors and BS are initially synchronized. BS preserves the synchronization with all sensors by broadcasting its clock time at regular intervals.
- Sensors belong to only one CH. Cluster structure is modified at regular intervals.
- In a cluster, a CH generates a Time Division Multiple Access (TDMA) schedule for transmission of member sensors. Each sensor transmits its data to the CH in its assigned time slots and keeps its state in sleep mode during remained slots. The CHs aggregate data from member sensors and send the aggregated data to the BS using a fixed spreading code and Carrier Sense Multiple Access (CSMA). That is, a CH first sense the channel to see whether there is a transmission from a different CH or not. If the channel is occupied by any other transmission, it should wait to transmit its data. Otherwise, it sends its data to the BS using the BS spreading code. Note that each cluster employs a different spreading code for intra-cluster communication to minimize the inter-cluster interference.
- Each sensor can vary the amount of transmit power, and support different Medium Access Control (MAC) protocols.
- An attacker randomly selects a sensor as a compromise target regardless of its role.

A cluster is different from the neighbors of a node and is created by the comparison of a specific criterion(s) among neighbor nodes. As a result, a neighbor of a CH may not be a member of the CH and any two neighbor nodes may be served by different CHs. Even though it is possible to use multi hop communication between a sensor and its CH, it is more vulnerable to attacks than the TDMA-based one hop communication. This is because intermediate nodes on the route to a CH can overhear and even modify the packet when they are compromised. So, we adopted the TDMA-based one hop communication in a cluster and all members are directly connected to its CH. Fig. 2 shows the communication model of the clustered sensor networks.

In our system, the BS is directly connected to infrastructure networks and situated in a very safe position. On the contrary, sensors are deployed in an unattended or unprotected environment so that they are likely to be targets of capture attacks. If some sensors are captured, their keys are exposed to attackers, and the attackers share the keys with other attackers using a covert channel. We assume two aims of attackers.
- The primary aim of attackers is to obtain sensor reading of other sensors.
- The second aim is to send bogus messages to BS without detection.

An intuitive example of above threat model is a military surveillance network. Sensors recognize the invasion of enemy troops, and inform the invasion to the headquarters. Some compromised sensors also recognize the invasion, but they do not give the alarm to the headquarters in order to hide the invasion. They rather inform a peaceful condition to the BS. If all of the CHs are compromised, the attackers can completely cheat the headquarters by forging sensed data from all sensors. Consequently, the invasion of the enemy troops is thoroughly hidden and the troops can give a fatal blow to the headquarters.

Generally, key renewal depends on the detection of compromised sensors. If there are no detection mechanisms or the detection is incorrect, the key renewal is functionally useless. So, the design and implementation of an Intrusion Detection System (IDS) is very important with respect to key renewal. However, this issue is beyond the scope of this paper. In recent years, the issue has been dealt in a lot of literatures including [15]. Here, we just make two additional assumptions.
- A CH can detect all compromised sensors in its cluster.
- BS can detect all compromised CHs.
Our scheme consists of four steps: pairwise key generation, cluster organization, CH-BS key agreement, and resynchronization and compromise report. The pairwise key generation step is invoked only one time at the network boot-up time. On the other hand, other steps are repeated in a periodical manner until the extinction of the network. Fig. 3 shows the procedure of the proposed scheme.

B. Pairwise Key Generation

In cluster based sensor networks, two kinds of keys are required for securing the communications from sensors to BS. One kind is used for securing the communications from sensors to their CH. These keys are called as intra-cluster keys. The other kind is used for securing the communications from CHs to BS. These keys are called as CH-BS keys. In the group key renewal schemes, there is only one intra-cluster key in a cluster, so called group key. Therefore, they are very vulnerable to compromise of a sensor, as described in Section II. We adopted a different approach where all sensors pre-establish pairwise keys with neighbors and use them for intra-cluster communication keys.

To generate the pairwise keys, each sensor invokes a key establishment procedure after deployment. Because they are initially synchronized, all sensors start the key establishment at the same time. Any two neighboring sensors agree a pairwise key using the mixture of a key pre-distribution method [9] and an ID-based method [16]. That is, a sensor establishes a pairwise key with a neighbor that shares common assigned keys by using the common keys [9]. If any two sensors share no common administrative keys, then they establish a pairwise key using the network-wide key and their IDs [16]. For the sake of comprehension, we list the notations used in this paper in Table I. Fig. 4 shows the pairwise key establishment when there are no common pre-assigned keys between any two sensors. In Fig. 4, sensors 1 and 2 broadcast a hello message including a nonce (that is, a random number) and their ID. If a sensor (e.g. 1) receives such a message from a higher ID sensor (e.g. 2), it ignores the message. Otherwise, it generates its master key, and employs its master key and the sender’s ID to generate the pairwise key. When the sensor 2 receives a hello message from sensor 1, it computes its master key \( MK_1 \) and the pairwise key \( PK_1 \) using its master key and sensor 1’s ID. Then it generates a Message Authentication Code (MAC) value for 1’s nonce and its ID using \( MK_1 \), and responds to sensor 1 with the MAC and its ID. Receiving the response, sensor 1 generates 2’s master key \( MK_2 \), and verifies the 2’s legality by verifying the MAC value. Finally, sensor 1 generates the pairwise key shared with sensor 2 \( PK_2 \). During the pairwise key establishment step, all nodes are assumed to be trustworthy and behave. Besides, the pairwise key generation is completed in a very short time so that an attacker cannot compromise a sensor in such a short moment. After the generation of pairwise keys, all sensors remove all pre-assigned keys and the network-wide key from its memory for a security reason. This is because these keys are basic materials supporting our key renewals. If these keys are exposed to an attacker, the attacker immediately gets a large part of pairwise keys in the network, and can debilitate the key renewal work.

C. Cluster Organization

After intra-cluster key generation, each sensor invokes a cluster organization procedure. The cluster organization procedure is invoked regularly using a timer. Note that the timer reset interval should be large enough to cover the CH-BS key agreement and communications from sensors to BS. This periodic cluster reorganization brings two advantages. First, it prevents the compromised sensors from joining the network operation by rejecting all messages from them during the cluster organization. All sensors recognize the blacklist of sensors by receiving the list of compromised sensors from BS every key renewal period. Note that this list is erased from the memory of sensors after the cluster organization. Second, CHs consume much more energy than member sensors because it regularly performs a long haul transmission to the BS. If some sensors play such roles continuously, they rapidly exhaust all energy so that network lifetime will be shortened accordingly. Our scheme avoids the problem by changing the CH role sensors periodically. Each sensor changes its state among the following three states and consequently makes a new cluster structure.

- INITIAL: Each node has this state after the network deployment and timer expiration. A node with this state broadcasts its ID and residual energy after setting its timer. An INITIAL node changes its state to MEMBER after the broadcast.
MEMBER: If a MEMBER node receives an advertisement message from a CH, it broadcasts a message which informs the CH of its affiliation. A MEMBER node which does not determine its role keeps receiving the message from neighbors until it can determine its role. If it detects that all neighboring MEMBERs with higher priority (that is, highest energy level or lowest ID under the same energy level) joins other clusters, it transits to CH state. Otherwise, it waits for other messages to determine its role.

Fig. 5 shows the example of node eviction and CH election in our scheme. Fig. 5(a) shows a beginning time of a key renewal period, and Fig. 5(b) shows the result of cluster organization in the network. Every beginning time of a key renewal period each node broadcasts a hello message including its ID and residual energy. Through this message, each node identifies the priority relationship between neighbors and they use it in order to determine its role. For example, node 4 recognizes that it is most eligible to be a CH because its residual energy is high and its ID is lower than node 9. In case of nodes 6, it cannot determine its role immediately because it has two higher energy nodes (that is, 1 and 5) and one lower ID node (that is 2). Like node 6, all other nodes have one or more nodes with a higher priority. So, they have to delay their role decision until their higher priority nodes declare as a CH or affiliate to another CH. First, node 4 declares itself as a CH. Because the nodes 1 and 5 affiliate to the cluster of CH 4, node 2 becomes the highest priority node among their neighbors. Therefore, node 2 declares itself as a CH, and nodes 6, 7, and 10 join the cluster of CH 2. Notice that nodes 1, 3, and 5 do not join the cluster of CH 2 because they have already joined the cluster of CH 4.

At this time, some compromised nodes may try to join the network. Because all nodes recognize all compromised nodes through the broadcast of the BS, they can evict them by refusing to communicate with them. This eviction is very important because the compromised nodes can declare itself as a CH. For example, in Fig. 5(a), even though the malicious nodes 22 and 27 send a hello message including its ID and a high residual energy, the receiving nodes 2, 3, 10, 4, 5, and 9 do not respond to the messages because the sender of the messages are known to be compromised.

D. CH-BS Key Agreement

After the cluster organization is completed, each CH agrees a key with BS. First, each CH generates a CH-BS key by XORing all pairwise keys shared with members. Then, it informs the BS of its members to make an agreement. The length of the list can be greatly reduced by configuring the list as a bit array. That is, a bit position which is assigned to a member is settled to “1.” Otherwise bits are all settled to “0.” The BS has already known the network-wide key and pre-assigned keys of the CH and the members because it is the dealer of those keys. So, the BS can easily make the same CH-BS key using the cluster membership information only. Note that there is no exchange of keying materials between the CH and BS. Besides, the CH-BS keys are changed periodically. This is because the cluster membership changes periodically owing to the cluster organization.

Next, each CH broadcasts a TDMA transmission schedule for its members. Upon receiving the schedule, each sensor knows when it transmits its reading and when it remains in sleep state. That is, sensors transmit their readings only in their allowed time slots.

E. Resynchronization and Compromise Report

Each CH recognizes the compromised sensors, and appends the list of the compromised sensors to aggregated data which is sent to the BS. The BS recognizes the compromised CHs, and holds the list of all compromised sensors to advertise it to sensors. When its timer expires, it broadcasts its clock time and the list of compromised sensors. The list of compromised sensors is configured using a bit array where a bit ‘1’ indicates the compromise of a sensor assigned to the bit position. This advertisement message makes the sensors to resynchronize their timer with the BS and recognize the all compromised sensors. Then, pure sensors make a secure cluster structure in the next cluster organization time by refusing to communicate with the blacklisted nodes. After the cluster reorganization, sensors erase the blacklist to save their memory space.

F. Scalability Consideration

In any sensor networks, sensors should be replaced at times because they are disabled by battery exhaustion and sometimes broken by external impacts. In this case, newly deployed sensors should do some pre-work, that is pairwise key establishment, to join the operation of our scheme. Because the existing nodes have already erased the network-wide key and pre-assigned keys, the pairwise key establishment method described in the subsection B cannot be reused. We make two additional assumptions to describe the pairwise key establishment between new joining nodes and the existing nodes. First, each sensor knows the generation they belong to, and is assigned a
Because there are totally attackers randomly select the targets to compromise. 

employed in this section. As we commented in Section III, using probability theory. Table II describes the symbols

nodes, it can easily join the operation of our scheme. Once a new joining node established pairwise keys with the existing role of an existing node in this procedure. If any two sensors are both in the new generation, they compare their IDs with each other. If a node receives a Hello message from a smaller ID node, it can easily join the operation of our scheme.

Fig. 6. Pairwise key establishment between different generations

First, a new joining node 9 sends a Hello message including its ID, a random number, and its generation in Fig. 6. If an existing node 2 receives the message, it responds to the node 9 without exception. The existing node 2 first extracts the master key for the new generation and generates the pairwise key using this master key and the node 9’s ID. Then, the node 2 sends its ID, the pairwise key encrypted with the master key for the new generation, and a MAC value for proving its legality to the node 9. The node 9 generates the node 2’s master key using its network-wide key and the ID of the existing node. Then the node 9 generates the pairwise key using its ID and the existing node’s master key. Finally, the node 9 compares the generated pairwise key to the decrypted pairwise key. If they are the same keys, then the node 9 employs the generated key as the pairwise key for the decryption of the node 2. If any two sensors are both in the new generation, they compare their IDs with each other. If a node receives a Hello message from a smaller ID node, it repeats the above procedure. Otherwise, it discards the received packet. Note that a smaller ID node plays the role of a new joining node and the other party plays the role of an existing node in this procedure. Once a new joining node established pairwise keys with the existing nodes, it can easily join the operation of our scheme.

IV. SECURITY ANALYSES

In this section, we analyze the security of our scheme using probability theory. Table II describes the symbols employed in this section. As we commented in Section III, attackers randomly select the targets to compromise. Because there are totally c compromised nodes, the probability \( P_j \) can be represented by the following.

\[
P_j = \frac{c!}{(c-j)!j!} \times \frac{(n-1)^{c-j}}{n^2} \quad \text{for} \quad j \leq |g| - 1.
\]

\[
P_{\text{member}}^i = \frac{1}{|g|} \times \frac{c!}{(c-|g|+1)!} \times \frac{(n-1)^{c-|g|}}{n^2}
\]

\[
\text{for} \quad c \geq |g| - 1.
\]

In a cluster, because there is only one CH and some members, the probability \( P_{\text{member}} \) and the probability \( P_{\text{CH}} \) are represented by (2) and (3).

\[
P_{\text{cluster}}^i = P_{\text{CH}}^i + P_{\text{member}}^i.
\]

In our scheme, attackers can illegally obtain data from sensors only through compromised sensors because all of the sensors employ different keys in a cluster. Therefore, \( P_{\text{cluster}}^i \) can be computed by (4). Because there are \( n \) clusters in the network, the probability \( P_{\text{network}} \) can be represented by (5).

A. Extension of Neighbor Radius(Cluster Radius)

If the neighbor radius is extended, \( P_{\text{member}} \) greatly decreases due to the increase of \( |g| \). Moreover, if the neighbor radius is extended and the number of compromised nodes is reasonably large, \( P_{\text{CH}} \) also decreases due to the increase of \( j \) value. That is, the increase of the neighbor radius causes the decrease of the probability \( P_{\text{member}} \). In other words, our scheme enhances the security when the neighbor radius (that is, cluster radius) is extended. However, the extended radius increases the energy consumption of the sensors because more sensors are involved in the cluster head election in a cluster.

B. Extension of Cluster Timer Interval

If the cluster timer’s interval gets longer, the value of \( c \) increases. This increase makes \( P_{\text{CH}} \) and \( P_{\text{member}} \) greater. That is, if the cluster timer’s interval is lengthened, the

---

**TABLE II**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>Number of clusters</td>
</tr>
<tr>
<td>( g_i )</td>
<td>i-th cluster</td>
</tr>
<tr>
<td>(</td>
<td>g</td>
</tr>
<tr>
<td>( c )</td>
<td>Number of compromised nodes</td>
</tr>
<tr>
<td>( P_j )</td>
<td>Probability that j sensors in a cluster are compromised</td>
</tr>
<tr>
<td>( P_{\text{member}} )</td>
<td>Probability that all members(except CH) in a cluster are compromised</td>
</tr>
<tr>
<td>( P_{\text{CH}} )</td>
<td>Probability that the CH in a cluster i is compromised</td>
</tr>
<tr>
<td>( P_{\text{cluster}}^i )</td>
<td>Probability that a cluster i is compromised</td>
</tr>
<tr>
<td>( P_{\text{network}} )</td>
<td>Probability that the entire network is compromised</td>
</tr>
</tbody>
</table>

\[
P_{\text{CH}}^i = \sum_{j=|g|}^{c} \frac{c!}{(c-j)!j!} \times \frac{(n-1)^{c-j}}{n^2} \times \frac{(n-1)^{|g|}}{n^2}.
\]

\[
P_{\text{network}} = \prod_{i=1}^{n} P_{\text{cluster}}^i.
\]
security of our scheme is weakened. On the other hand, the lengthened interval reduces the energy consumption of the sensors because the frequency of key renewal invocation decreases.

V. SIMULATION RESULTS

We built the simulation environment (ns-2 version 2.27) to evaluate our scheme in terms of security and efficiency. We deployed 100 sensors on random positions in a 1000meters×1000meters area. The BS was deployed in the position of (50meters, 175meters). We adopted the energy consumption model of [17]. We compared our scheme with SHELL which is the most representative one of group key renewal schemes. We executed both scheme 30 times per each number of compromised nodes to average the simulation results. Then we executed our scheme 30 more times for each neighbor radius and averaged the results. Table III lists the simulation parameters and their values employed for the simulations. In the EBS (Exclusion Basis System) parameter, $k+m$ refers to the size of key pool in a CH and $k$ refers to the size of the key ring in a member sensor. We developed five main metrics for the comparison.

- Exposure rate: rate that shows how many sensor readings are exposed to compromised sensors (that is attackers). It represents the confidentiality in communications from sensors to BS.
- Fabrication rate: rate that shows how many sensor readings are modified by compromised sensors and delivered to BS. It represents the integrity in communications from sensors to BS.
- Activity time of attackers: the average time during which compromised nodes do malicious work until they are evicted or extinguished from the network. This metric is used to measure the resiliency of a key renewal scheme.
- Energy consumption rate: rate that shows what amount of energy is consumed for key renewal process. This rate is computed through dividing total energy consumption for key renewals by total energy consumption during network lifetime. It represents the energy-efficiency of a key renewal scheme.
- Network lifetime: activity time of the network. This metric is used to show the impact of a key renewal scheme on the availability of the network.

Fig. 7 shows the exposure rate of the sensed data as the number of compromised nodes increases. The increase in the number of compromised nodes increases the exposure rate in both schemes.

Because SHELL employs only one group key in a cluster, one compromised sensor can obtain all communications within its hearing range. As the number of compromised nodes increases, the amount of exposed data also increases. If a compromised CH is detected in SHELL, the sensors that belonged to the compromised CH are redistributed to pure CHs. This increases the density of the sensors in a cluster. Therefore, compromised nodes in a cluster can illegally acquire data from more sensors.

In our scheme, because sensors send their data to the CH using the pairwise keys, it is much less affected by the compromise of sensors. That is, attackers cannot obtain data from other sensors through a compromised node. In addition, when a CH is compromised, our scheme elects new CHs among the pure sensors so that it does not increase the density of sensors in a cluster.

Fig. 8 shows the fabrication rate of sensed data as the number of compromised nodes increases. In SHELL, as

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation time</td>
<td>3600 seconds</td>
<td></td>
</tr>
<tr>
<td>Simulation area</td>
<td>1000m.×1000m.</td>
<td>Common</td>
</tr>
<tr>
<td>Initial energy of sensors</td>
<td>10 Joules/battery</td>
<td></td>
</tr>
<tr>
<td>Bandwidth</td>
<td>1 Mbps</td>
<td></td>
</tr>
<tr>
<td>Data size</td>
<td>500 bytes</td>
<td></td>
</tr>
<tr>
<td>Packet header size</td>
<td>25 bytes</td>
<td></td>
</tr>
<tr>
<td>Number of sensors</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Number of compromised sensors</td>
<td>10–50</td>
<td></td>
</tr>
<tr>
<td>Compromise time of sensors</td>
<td>Random(0–900 seconds)</td>
<td>SHELL</td>
</tr>
<tr>
<td>Number of clusters(CHs)</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Number of compromised CHs</td>
<td>2–4</td>
<td></td>
</tr>
<tr>
<td>Number of assigned keys($k$)</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Number of non-assigned keys($m$)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Detection and renewal interval</td>
<td>20 seconds</td>
<td></td>
</tr>
<tr>
<td>Neighbor radius</td>
<td>30 meters</td>
<td></td>
</tr>
<tr>
<td>Timer reset interval</td>
<td>60, 120, 180 seconds</td>
<td>Our scheme</td>
</tr>
</tbody>
</table>

Figure 7. Exposure rate vs. compromised nodes

© 2010 ACADEMY PUBLISHER
the number of compromised nodes increases, the fabrication rate increases accordingly. This is because SHELL evicts the compromised sensors very passively. SHELL first renews the administrative keys known to the compromised nodes and then renews the group key using the renewed administrative keys. This eviction scheme has no effect on the security of the network if all of the administrative keys are known to the attackers. In this case, attackers can keep fabricating the data of the compromised sensors. Our scheme guarantees the removal of the compromised nodes using a cluster reorganization process that excludes the compromised nodes. This makes the slope of the fabrication rate much gentler even if the number of compromised nodes increases.

Fig. 9 shows the activity time of the compromised nodes (that is, attackers) as the number of compromised nodes increases. As shown in Fig. 9, the compromised nodes in SHELL survive the eviction process for a long time and continue to exhibit malicious behavior. Note that the activity time of the compromised nodes decreases with the increase in the number of compromised CHs. This is because the BS instructs the sensors and CHs to reorganize the cluster structure when it detects a compromised CH. Because the BS broadcasts the compromised CHs and sensors at the beginning of the cluster reorganization, they are evicted from the network. Consequently, the increase in the number of the compromised CHs increases the frequency of the cluster reorganization, and the activity time of the compromised nodes decreases.

Our scheme evicts the compromised nodes at regular intervals through periodic cluster reorganization. This makes the activity time of the compromised nodes very short. Although the increase in the cluster timer interval makes the activity time slightly longer, the lengthened times are still much shorter than those of SHELL.

Fig. 10 shows the number of nodes evicted by the IDS which operates together with the key renewal scheme as the number of compromised nodes increases. In SHELL, the evicted nodes increases as the number of compromised CHs increases due to frequent reorganization of clusters. That is, the IDS evicts the compromised sensors by redistributing the pure sensors to pure CHs excluding the compromised sensors. Our scheme more frequently evicts the compromised sensors by invoking the cluster reorganization process periodically. That is, the pure sensors evict the...
compromised sensors by refusing the communication with the compromised sensors during the cluster reorganization process. Therefore, our scheme shows a better performance than SHELL as shown in Fig. 10.

Fig. 11 shows the energy consumption rate as the number of compromised nodes increases. SHELL has three heavy energy-consuming procedures in the key renewal operation. They are “cluster organization,” “administrative key renewal and distribution,” and “group key renewal and distribution using administrative keys.” This is because these procedures require many sensors to be involved in the communication and the computation of the rekeying operation. If there are only compromised sensors (that is, no compromised CHs), only the last two procedures are performed. However, if at least one CH is compromised, then the sensors should first engage in the cluster organization procedure and this requires extra communication overhead. Therefore, the sensors consume more energy, if the number of compromised CHs increases. Our scheme consists of four procedures; pairwise key generation, cluster organization, CH-BS key agreement, and resynchronization and compromise report. However, in our scheme, the cluster organization procedure is the only procedure that uses substantial energy. The pairwise key generation is performed only once after deployment. Moreover, the CH-BS key agreement makes only a small number of sensors join the procedure. Finally, the resynchronization and the compromise report make sensors consume a very small amount of energy to receive a message from the BS. As a result, our scheme reduces the energy consumption for key renewal as shown in Fig. 11. When only a small number of nodes are compromised (<20) and the cluster timer interval is short (60 seconds), our scheme consumes more energy compared to SHELL. However, as the number of compromised nodes increases, our scheme reduces the energy consumption significantly. This is because our scheme evicts the compromised nodes expeditiously while SHELL revives most of them when the density of the compromised nodes is high.

Fig. 12 shows the number of nodes excluded by energy exhaustion as the number of compromised nodes increases. In SHELL, sensors consume much more energy for key renewals than our scheme so that they exhaust their energy quickly. Our scheme consumes much less energy for key renewals than SHELL as shown in Fig. 11. Therefore, the number of sensors which is extinguished from network decreases significantly. Our scheme shows a better performance than SHELL excluding the case of four compromised CHs.

Fig. 13 shows how the key renewal schemes affect the network lifetime as the number of compromised nodes increases. Both schemes evict the compromised nodes. The eviction of the compromised nodes reduces the number of active nodes in the network. Because the remaining active nodes continue to consume their energy for key renewal and transmission of their data, the number of active nodes continuously decreases with the lapse of time. If the number of active nodes is equal to the number of CHs, the network is extinguished.

In SHELL, whenever the BS detects the compromised CHs, it evicts them from the network through cluster reorganization. Therefore, an increase in the number of compromised CHs causes the number of evicted nodes to increase as well. Nevertheless, the network lifetime lengthens as shown in Fig. 13. This is because the network lifetime is dominated not by the number of evicted nodes but by the number of remaining pure CHs. That is, if the number of compromised CHs is small (that is, two), then the sensors are redistributed to the remaining pure CHs. In this case, all sensors are properly distributed to three remaining CHs so that the number of sensors served by a CH is small. This distribution makes the transmission schedule in a cluster short, and sensors frequently transmit their data to the CH. That is, because sensors consume large amounts of energy, the number of active nodes rapidly decreases. On the other hand, if only one pure CH (that is, four compromised CHs) exists in a network, then the CH should take charge of all of the sensors. This makes the transmission schedule in the cluster very long and the transmission frequency of the
sensors decreases greatly. This slows the energy consumption of the sensors and the number of active nodes decreases very slowly.

Our scheme extends the network lifetime as the cluster timer interval increases. This is because the frequency of key renewal decreases due to the lengthened expiration time of the cluster timer. However, our scheme still makes the network lifetime shorter than SHELL with four compromised CHs. Because our scheme generates small sized clusters, the transmission schedules of the clusters are much shorter than those of SHELL. This short schedule requires much more transmissions from the sensors and shortens the lifetime of the sensors, thereby reducing the network lifetime. Nevertheless, our scheme lengthens the network lifetime compared to SHELL when the number of compromised CHs is less than four.

VI. EFFECT OF NEIGHBOR RADIUS EXTENSION

We examined the effects of neighbor radius extension on our scheme through additional experiments. We measured the variance of the metrics described in Section V (excluding activity time of attackers) with the increase of neighbor radius. In these additional simulations, the number of compromised nodes is fixed to 30. If the number of member sensors is equal to the number of CHs, the network is extinguished and the time is recorded as the lifetime. Because all sensors elect a CH among all its neighbors, the neighbor radius can be called as cluster radius. Hereafter, we called the neighbor radius as cluster radius.

Fig. 14 shows the variation of the exposure rate as the cluster radius increases. If the cluster radius increases, the number of CHs decreases dramatically. Because attackers choose the compromise targets randomly, the probability that a CH is compromised decreases. If the number of compromised CHs decreases, illegal data acquisition using these CHs also decreases.

Fig. 15 shows the variation of the fabrication rate as cluster radius increases. Like the exposure rate, the fabrication rate is much more affected by the fabrications using compromised CHs than those using compromised member sensors. Therefore, the fabrication rate shows a result that is very similar to that for the exposure rate.

Fig. 16 shows the energy consumption rate of our scheme as the cluster radius increases. If the cluster radius increases, the number of nodes that join the cluster organization greatly increases, and energy consumption for key renewals increases accordingly. That is, an increased cluster radius enhances security but sacrifices the energy consumption rate.

Fig. 17 shows the variation of the network lifetime as the cluster radius increases. The increase in the cluster radius reduces the number of CHs and increases the number of sensors belonging to a cluster. This makes the transmission schedule of the clusters very long, and the transmission frequency of each sensor is reduced. Consequently, energy consumption at each sensor is reduced and the network lifetime is lengthened.

Through the above simulation results, we can conclude the following. Considering security and efficiency at the same time, an appropriate value of cluster radius is 60 meters. This is because this radius extends the network lifetime to the end of the simulation time as shown in Fig. 17. Among all radii that guarantee the same lifetime (>=
60 m.), this radius minimizes the energy consumption rate as shown in Fig. 16. Of course, this radius slightly impairs confidentiality and integrity compared to further extended ranges as shown in Fig. 14 and Fig. 15. However, these values are still small compared to those of SHELL as shown in Fig. 7 and Fig. 8. In addition, the above simulations reveal another important consideration. If the cluster timer interval is 120 seconds, the radius of 60 meters maximizes the network lifetime without sacrificing confidentiality and integrity as shown in Fig. 14 and Fig. 15.

VII. DETECTION OF COMPROMISED NODES

As described in Section III, a key renewal scheme highly depends on a well-behaved IDS. The incorrectness and malfunction of an IDS sometimes put innocent nodes into the blacklist or make compromised nodes slip out of surveillance of the IDS. We discuss how an IDS can detect the compromised nodes in our scheme.

Karlof et al. pointed out various threats which are available on sensor networks and proposed their countermeasures in [18]. In a clustered sensor network, a number of attacks such as sybil attack, black-hole attack, selective forwarding attack, and DoS attack are available. Among them, the hello flood attack and the sybil attack are hardly detected by an IDS. In our scheme, any two sensors within transmission range shared a pairwise key at network boot-up time and only one hop clusters are formed during network operation. This prevents an attacker from invoking the hello flood attack and the sybil attack. That is, a pure sensor can recognize an attacker which invokes a hello flood attack because the attacker is not a node with which established a pairwise key at network boot-up time. Then, the pure sensor escapes from the hello flood attack by rejecting the communication with the attacker. In the same manner, a pure sensor refuses to communicate with an attacker which invokes a sybil attack. The rest of attacks rely on the packet transmission and reception, so they can be easily identified by an IDS. To detect these attacks in a clustered network, we need to alleviate some constraints of TDMA based communication. This is because some sensors should monitor a sensor’s behavior in their cluster. So, they do not change into the sleep mode in unassigned slots. To grasp a quick comprehension of the detection process, we describe it through an illustrative example. As shown in Fig. 5(b), in a time slot of node 1, nodes 3, 4 and 5 are assigned to monitor node 1’s behavior. In the same manner, CH 4 can be monitored by nodes 1, 3, 5, 8, and 9. If node 1 does not forward its data, nodes 3, 4, and 5 increase the suspicion counter by one. If the counter exceeds a threshold, they create an alarm message including the malicious node and transmit the message with their own data to all other members. If the CH 4 does not transmit the aggregated data (this attack can be called as DoS attack), its all members do the same work as described just before. Sometimes, a compromised CH can intentionally include innocent nodes to the list of malicious nodes to hide its misbehavior. However, this blacklist fabrication is detected by all well-behaved members and becomes null by their cooperation. That is, they elect a proxy CH which collects the complaints from well-behaved members. Then the proxy CH reports the number of complaints to the BS so that the BS can evict the compromised CH.

VIII. CONCLUSION

In this paper, we proposed a lightweight key renewal scheme which realizes the renewal by not administrative keys but cluster reorganization. Main contribution of our scheme is that it provides an explicit way to evict compromised sensors and a way to employ and renew distinct intra-cluster keys in a cluster. Besides, our scheme offers a novel way to renew CH-BS keys. We compared our scheme with a group key renewal scheme using simulations in terms of security and energy-efficiency. The simulation results show that our scheme is more tolerant to sensor compromise than the group key renewal scheme. Besides, they show that our scheme saves the energy and extends the network lifetime more than the group key renewal scheme.

ACKNOWLEDGMENT

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No. R01-2008-000-20062-0)

REFERENCES


Gicheol Wang received the B.S. degree from Gwangju University, Gwangju, Korea, in 1997, and the M.S. degree from Mokpo National University, Mokpo, Korea, in 2000, in computer science and statistics. He received Ph. D degree in computer science and statistics from Chonbuk National University, Jeonju, Korea, in 2005.

He worked for CAIIT(Center for Advanced Image and Information Technology) at Chonbuk National University, Jeonju, Korea, as a Postdoctoral Research Fellow from Jan. 2006 to Dec. 2007, for the Research Center for Ubiquitous Information Appliances at Chonnam National University, Gwangju, Korea, as a Postdoctoral Research Fellow from Jan. 2008 to Dec. 2008. From Jan. 2009, he joined to the Div. of Supercomputing Center at KISTI(Korea Institute of Science and Technology Informations), Daejeon, Korea, and he is currently serving as a senior research scientist. His current research interests include mobile computing, computer communication, network security, sensor networks, and network measurement.

Dr. Gicheol Wang is a member of IEEE, KIPS, IEEK, KSII.

Seunghae Kim was born in Yeosu, South Korea, Aug 28, 1973. He received the B.S. degree from Hankam University, Daejeon, Korea, in Information and Communication Engineering, 1997, and the M.S. degree from Chonbuk University, Chonju, in Computer Science, 2003. He received Ph. D degree in Information Security from Chonbuk University, Chonju, in 2008.

He worked for ETRI(Electronics and Telecommunications Research Institute), Daejeon, Korea, as a Senior Member of KREONET(Korea’s National Research Network) from Sep. 1996 to 1998. From 1999, he joined to the Division of Supercomputing Center at KISTI(Korea Institute of Science and Technology Informations), Daejeon, Korea, and he is currently serving as a Senior Researcher and leader of KREONET team. His current research interests include optical network, computer communication, network security, sensor networks, and network measurement.

Dr. Kim is a member of IEEE, KIPS, IEEK, KSII.

Deokjai Choi received the B.S. degree from Seoul National University, Korea, in 1982, and the M.S. degree from KAIST (Korea Advanced Institute of Science and Technology), Korea, in 1984, in computer sciences. He received Ph. D degree in computer science and telecommunications from University of Missouri-Kansas City, USA, in 1995.

He worked for Chonnam Nat’l University, Gwangju, Korea, as a faculty member since 1986 until now. His current research interests include mobile computing, mobile agent, sensor networks, and ad hoc networks.

Prof. Kang is a member of KIPS and KMMS.

Daewook Kang received the B.S. degree from Chonnam National University, Korea, in 1982, and the M.S. degree from KAIST (Korea Advanced Institute of Science and Technology), Korea, in 1984, in computer sciences. He received Ph. D degree in computer science and telecommunications from University of Missouri-Kansas City, USA, in 1995.

He worked for Chonnam Nat’l University, Gwangju, Korea, as a faculty member since 1984 until now. His current research interests include mobile computing, mobile agent, sensor networks, and ad hoc networks.

Prof. Kang is a member of IEEE, IEEK, KIIEE, KSII, and IEEK.

Gihwan Cho received the B.S. degree from Chonnam University, Gwangju, Korea, in 1985, and the M.S. degree from Seoul National University, Seoul, Korea, in 1987, in both in computer science and statistics. He received Ph. D degree in computer science from University of Newcastle Upon Tyne, England, in 1996.

He worked for ETRI(Electronics and Telecommunications Research Institute), Daejeon, Korea, as a Senior Member of Technical Staff from Sep. 1987 to Aug. 1997, for the Dept. of Computer Science at Mokpo National University, Mokpo, Korea, as a Full Time Lecture from Sep. 1997 to Feb. 1999. From Mar. 1999, he joined to the Division of Computer Science and Engineering at Chonbuk National University, Chonju, Korea, and he is currently serving as a professor and chairman of the division. His current research interests include mobile computing, computer communication, security of wireless networks, sensor networks, and distributed computing system.

Prof. Cho is a member of IEEE, KIIEE, KIPS, KMMS, KSII.