Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2011, Article ID 296704, 10 pages
doi:10.1155/2011/296704

Research Article

A Forward Authentication Key Management Scheme for Heterogeneous Sensor Networks

Jen-Yan Huang, I-En Liao, and Hao-Wen Tang

Department of Computer Science and Engineering, National Chung Hsing University, Taichung 402, Taiwan

Correspondence should be addressed to I-En Liao, ieliao@nchu.edu.tw

Received 1 June 2010; Revised 13 September 2010; Accepted 23 October 2010

Academic Editor: Damien Sauveron

Key encryption technology is a basic technique for protecting the secrecy of transmitted data among sensor nodes in wireless sensor networks. However, sensor nodes are inherently limited by insufficient hardware resources such as memory capacity and battery lifetime. As a result, few current key management schemes are appropriate for wireless sensor networks. This paper proposes a new key management method that uses dynamic key management schemes for heterogeneous sensor networks. The proposed scheme loads a hash function into the base station, cluster heads, and sensor nodes. The cluster heads and sensor nodes then generate their own keychains to provide forward authentication in case of key changes, security breaches, key changes due to security breaches. The cluster heads and sensor nodes establish pairwise keys to ensure transmission secrecy. The proposed scheme decreases the number of keys required for sensor nodes and cluster heads and is robust to the following attacks: guessing attacks, replay attacks, man-in-the-middle attacks, node capture attacks, and denial-of-service attacks.

1. Introduction

Wireless sensor networks (WSNs) consist of many sensor nodes capable of wireless communication and data collection. In addition to sensor nodes, most WSNs include two other components, which are base station and cluster head.

WSNs are suitable for military applications, environmental monitoring, meteorological data collection, medical information monitoring, and so on. WSNs solve the wiring problem that traditional wired networks face. Wireless sensor nodes have the advantages of small size, easy deployment, and dynamic configuration.

Sensor nodes are limited by insufficient hardware resources, such as memory capacity, battery lifetime, and processor speed. The limitations of memory determine the amount of data to be stored, while battery lifetime determines the life of sensor nodes and slow processors cannot handle complex computations. These problems in turn will influence the efficiency of sensor networks.

Researchers have previously proposed some key management schemes for homogeneous sensor networks [1]. In this type of environment, all sensor nodes have the same characteristics, such as battery lifetime, computation power, and memory capacity. However, this scheme encounters the problems of low transmission speed, limited scalability, and a lack of fault tolerance [1]. Heterogeneous sensor networks (HSNs) can avoid these problems. In HSNs, which include several kinds of sensor nodes, different kinds of sensor nodes have different properties and transmission ranges.

This study proposes a key management system for a heterogeneous sensor network. The members of this network include a minority of powerful high-end sensors (H-sensors), which work as cluster heads, and a majority of low-end sensors (L-sensors). The high-end sensors have more memory, a wider transmission range, longer battery and greater fault tolerance. Low-end sensors represent general sensor nodes.

Regarding the security issues in the wireless sensor network, the encrypting scheme must not increase the load of sensor nodes. If sensor nodes need to perform complex computations for encryption, it would consume the energy of sensor nodes. Hence, the traditional encrypting and decrypting method is not suitable for wireless sensor networks.

In the proposed method, the L-sensors only store a little data at a time. Hence, they only require a little memory

to work quickly. H-sensors regularly replace the encrypting key based on the status of the cluster. At the same time, the L-sensors can determine if the new key is legal. This design requires fewer resources to achieve the security of sensor nodes in wireless sensor networks, while ensuring confidentiality, integrity, and availability.

Following this introduction, the structure of this paper is as follows. Section 2 reviews related work. Section 3 describes the proposed scheme. Section 4 provides the security analysis. Section 5 presents system analysis. Finally, Section 6 offers conclusions.

2. Related Work

This section discusses related research about the foundation of security mechanisms and key management schemes for wireless sensor networks.

2.1. Foundation of Security Mechanism. A typical WSN transmits data between nodes via radio. To protect the security of data transmission, a key cryptosystem can ensure the confidentiality, availability, and integrity of data.

2.1.1. Message Authentication Code. The message authentication code (MAC) [2] performs message authentication using the secret key shared by the sender and receiver. The receiver can verify the validity of messages with the MAC. The proposed scheme combines the encryption method and the MAC algorithm, as Figure 1 illustrates.

2.1.2. Hash Function. The proposed scheme is based on the one-way hash function [3, 4]. A hash value, generated by a hash function $H(X)$, is given by $h = H(X)$, in which X is a variable-length message and $H(X)$ is the hash value with a fixed length. The hash value is appended to the message, allowing the receiver to authenticate it; the hash function itself is not a secret. The hash function is the "fingerprint" of a file, a message, or other block of data.

2.2. Key Management Schemes for Wireless Sensor Networks. In wireless sensor networks, there are three methods of assigning keys: random, deterministic, and hybrid [1]. In the random method, the system randomly chooses several keys from the key pool and then loads them into sensor nodes to create the key-chain. The deterministic method uses dynamic computation to generate keys that can enhance the connection between sensor nodes. In addition, the system can update the key periodically through different situations. By updating the key, the system can isolate malicious nodes and maintain security. The hybrid method combines the advantages of these two methods.

Eschenauer and Gligor [5] proposed a random key predistribution scheme that focuses on symmetric encryption and decryption. To build the initial encrypting and decrypting key between sensor nodes, the system first generates a huge key pool. The sensor nodes then randomly choose several keys from the key pool and load them before deployment. The sensor nodes use these preloaded keys to generate pairwise keys, which create safe communication channels between neighboring nodes. This communication channel is called a key path, and it allowed sensor nodes to connect with other nodes in the environment. To protect the confidentiality of the key path, each key corresponds to only one index value. However, when an attacker finds the key, the sensor nodes immediately change the index value to update the key and select a new pairwise key.

Chan et al. [6] proposed a predistribution q-composite key method that allows two sensor nodes to set the pairwise key only when they share at least q public keys.

In an attempt to improve upon these two methods, Du et al. [7] proposed a key management method applicable to heterogeneous sensor networks. This approach uses a small number of sensor nodes that have superior performance to load more keys, increasing the probability of the shared keys.

Liu et al. [2] proposed the grid-based key predistribution scheme and the random subset assignment scheme. These methods can build the pairwise key between sensor nodes in the wireless sensor networks. Liu and Ning proposed a scheme [8] that has the great advantage of predicting coordinates in the sensor nodes and then distributing suitable keys in advance.

Li et al. [9] proposed a hexagonal grid key predistribution scheme that uses a hexagonal coordinate system and binary polynomial. Zhang et al. [10] proposed a method in which sensor nodes insert their own coordinates and IDs into the hash function and then generate pairwise keys to communicate with each other. This enhances the relationship between sensor nodes. However, this method lacks an authentication scheme between adjacent sensor nodes.

The researches [11–13] proposed the location-aware deployment model of keys predistribution scheme. This approach divides the environment into several square areas and randomly deploys the sensors in each area. The system can be aware of location of the sensor nodes according to the sensor node's ID.

Liu et al. [14] proposed a group-based keys predistribution scheme that divides the sensor nodes into groups and scatters them. After deployment, the sensor nodes may suffer from wind force or terrain condition, making it likely that in-group sensor nodes likely become neighbors. Finally, they modeled the deployment distribution as a Gaussian distribution. Building pairwise keys between in-group sensor nodes and cross-group sensor nodes offers several advantages. Hence, they built the pairwise key between sensor nodes in the same group using the in-group key predistribution method and used the cross-group key predistribution method to build the pairwise key between adjacent sensor nodes in the different groups.

Moharrum and Eltoweissy [15] compared the merits and faults of the dynamic key generation and static key generation methods. Based on this analysis, they proposed a new method called an exclusion basis system (EBS) based on the dynamic key management scheme. Eltoweissy et al. [16] proposed a localized combinatorial keying (LOCK) method that generates the dynamic key based on the EBS.

Perrig et al. presented two security protocols [17] for sensor networks, called SNEP and μTESLA hereafter. SNEP

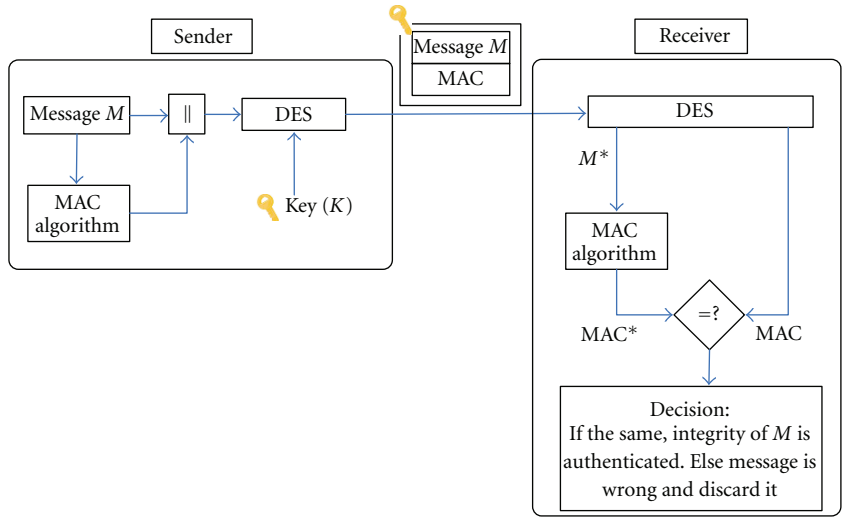


Figure 1: Message authentication code.

achieves data confidentiality and data authentication, while μTESLA ensures data integrity. In these structures, each sensor node shares the secret key with the base station. The base station functions as a trusted third party to keep and distribute the secret key. Younis et al. [18] and Jolly et al. [19] proposed a scheme much like the dynamic key generation model. The scheme can update and change the key through the certification authority (CA).

Chan and Perrig [20] proposed a protocol called peer intermediaries for key establishment (PIKE). In this approach, sensor nodes are trusted third parties and manage the key. Guorui et al. [21] proposed a group-based dynamic key management scheme. This system can update and change the key independently of the base station or cluster head. Cheng and Agrawal [22] proposed an effective method to build and manage the pairwise key. In the scheme, the system generates a two-dimensional key matrix, and each sensor node randomly stores one column and row of the key array from the matrix before deployment. After the sensor nodes are deployed, two adjacent sensor nodes can generate the pairwise key of each other.

Kausar et al. [23] proposed a hierarchical sensor network consisting of a small number of high-end sensors (H-sensor node) and a large number of low-end sensors (L-sensor node). The scheme is a scalable protocol for key management in the sensor networks to address the sensor nodes resource constraints, including computation, storage, and communication.

3. Proposed Scheme

This paper proposes a key-chain protocol for key management that is designed for heterogeneous sensor networks (HSNs). Each cluster head generates its own key-chain, which encrypts messages and communicates with the other sensor nodes in the cluster. Based on hierarchical clustering, each cluster consists of several sensor nodes and a cluster head. Several clusters and a base station form the heterogeneous sensor networks.

There are two types of sensors in hierarchical clustering HSNs: a small number of powerful high-end sensors (H-sensors, the same as the cluster head) and a large number of low-end sensors (L-sensors, the same as the ordinary sensor node). The H-sensors are equipped with tamper-resistant hardware and have more memory and greater processing capability. They can communicate directly with the base station. The L-sensors are normal sensor nodes that are limited in terms of processing capability, power, and memory. L-sensors acquire data from the surrounding environment and forward the collected data to the H-sensors. The H-sensors can communicate directly with the base station; all the L-sensor packets are transmitted to the BS via the H-sensor. This approach assumes that the base station is trusted. Figure 2 shows the architecture of hierarchically clustered HSNs.

3.1. System Setup. This section discusses the initialization and authentication phases in HSNs, including setting up the key-chain and setting up pairwise keys for the L-sensor nodes.

The proposed system assumes the following five communication rules.

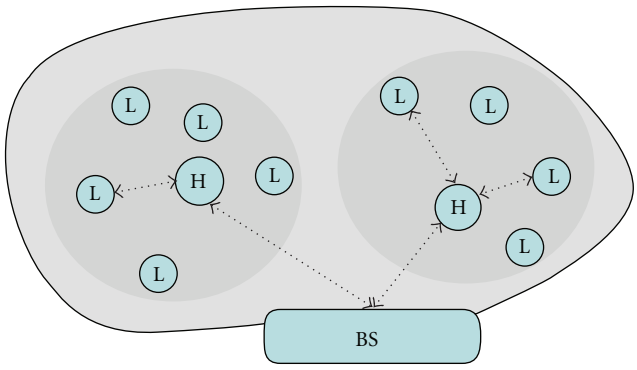
(1) H-sensors can directly communicate with the BS.

(2) The base station exchanges messages with L-sensors through H-sensors and vice versa.

(3) H-sensors can send messages to specific L-sensors in the cluster.

(4) H-sensors can broadcast messages to all L-sensors in the cluster.

(5) L-sensors must exchange the messages with each other through an H-sensor. In other words, L-sensors cannot directly exchange messages with each other. Hence, a compromised L-sensor cannot affect the other L-sensor in the cluster.



BS: Base station
H: H-sensor node
L: L-sensor node

Figure 2: Architecture of hierarchical clustering HSNs.

These communication rules are usually assumed for the hierarchical sensor networks such as SPINS [17], Gupta and Younis [24], and LEACH [25]. In this paper, these communication rules should be followed in order to avoid a compromised node infringing the other L-sensors and to prevent the attacks such as replay attacks or man-in-the-middle attacks.

3.1.1. Initialization Phase. The base station generates a key pool of size P before deployment of r L-sensors and q H-sensors, where $P \gg q$. The base station then chooses a unique key for each H-sensor, which is regarded as cluster key HK.

Before the deployment, the BS uses HK and random number R_S to generate a subkey $K_S = H(\mathrm{HK} \oplus R_S)$, and then uses K_S and $R_1 \sim R_n$ to generate a key-chain for each H-sensor as shown below:

$$\begin{aligned} K_{n-1} &= H(K_S \oplus R_n), \\ K_{n-2} &= H(K_{n-1} \oplus R_{n-1}), \\ &\vdots \\ K_1 &= H(K_2 \oplus R_2), \\ K_0 &= H(K_1 \oplus R_1). \end{aligned} \tag{1}$$

Hence, each H-sensor will obtain distinct key-chains, K_S, and random numbers $R_1 \sim R_n$ from the BS. H-sensor and L-sensor are stored with the same hash function $H(\cdot)$ and K_T, where K_T is a temporary session key for all H-sensors and L-sensors, and $K_T \neq HK$. All keys and parameters for each node will be passed from BS to sensor nodes through an offline secure channel.

H-sensors and L-sensors are randomly distributed in the environment. Each node is static and aware of its own location. H-sensors and L-sensors can use the protocol in [26] to evaluate the locations without GPS devices. Section 5 discusses the length n of the key-chain. To illustrate the system effectively, this study considers a single cluster. Table 1 presents the notation related to sensor nodes.

Table 1: The definition of the notations.

HK_j	The jth cluster key of H-sensor
$\mathrm{LK}_{i,j}$	The pairwise key between L-sensor i and H-sensor j
HID_j	The unique ID for H-sensor j
LID_i	The unique ID for L-sensor i
K_l	The lth key in the key-chain, where $1 \leq l \leq n$
K_T	A temporary session key for all H-sensors and L-sensors
RN_H	A random number generated by H-sensor
$\{\mathrm{MAC}(M) \mid\mid M\}_K$	The encryption of message M with MAC using the key K
$\{M\}_K$	The message M is encrypted by K
$H(\cdot)$	A collision-resistant cryptographic hash function $H : \{0,1\}^* \rightarrow \{0,1\}^{160}$
$\oplus$	XOR operator
$\mid\mid$	A concatenation operator

3.1.2. Authentication Phase. After all nodes are distributed in the environment, the H-sensors decide which nodes to connect with. To explain the environment, this paper focuses on describing the operations within one cluster.

(1) An H-sensor j broadcasts a hello message to all the neighboring L-sensors using the maximum power, where the hello message includes the H-sensor's ID HID_j. The location of the H-sensor j and a random number RN_H is encrypted by K_T. The format of hello message is as follows:

$$\mathrm{HID}_j \mid\mid \text{hello message} \mid\mid \text{Location of the H-sensor} \mid\mid \{\mathrm{RN}_H\}_{K_T}. \tag{2}$$

(2) The L-sensor i may receive one or more hello messages if no barricades are sheltering it. The L-sensor i chooses an H-sensor as its cluster head according to the distance and best signal strength of the message. In this environment, each L-sensor notes other H-sensors from which it receives the hello messages, and these H-sensors are recorded as backup cluster heads in case the chief cluster head is disabled. If the L-sensor i receives the message, it then takes its own LID_i and RN_H and generates a pairwise key $\mathrm{LK}_{i,j} = \{H(\mathrm{RN}_H \mid\mid \mathrm{LID}_i)\}_{K_T}$, replying to the H-sensor. The format of this response message is as follows:

$$\mathrm{HID}_j \mid\mid \text{response message} \mid\mid \text{Location of the L-sensor} \mid\mid \left\{\mathrm{MAC}\left(\mathrm{LK}_{i,j}\right) \mid\mid \mathrm{LID}_i\right\}_{K_T}. \tag{3}$$

Plain text can be used to deliver the HID_j in the message. Therefore, the receiver node can avoid decrypting the message, saving time and power.

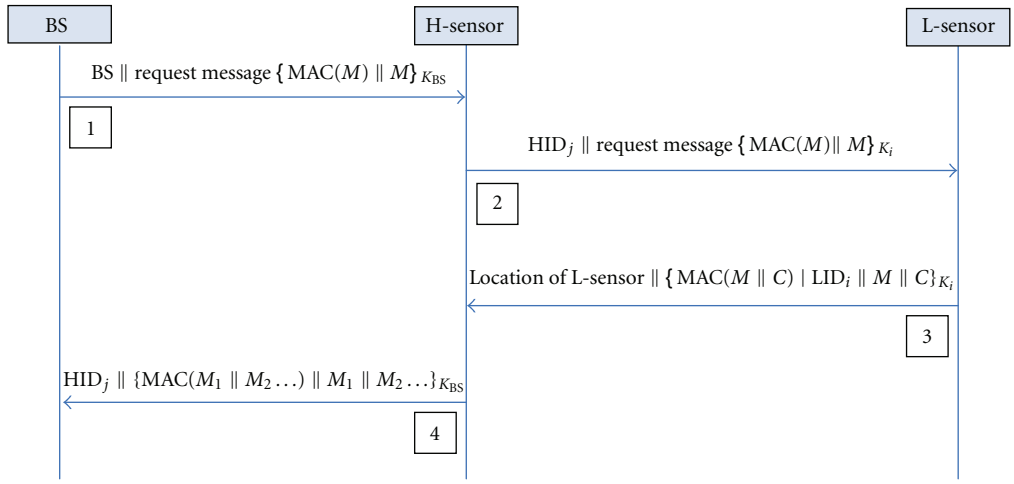


Figure 3: BS requires the data from the HSNs.

(3) After receiving the response message and LID_i of the L-sensor i, the H-sensor j generates pairwise key $\text{LK}^*_{i,j} = \{H(\text{RN}_H \;||\; \text{LID}_i)\}_{K_T}$. If the condition $\text{MAC}(\text{LK}^*_{i,j}) = \text{MAC}(\text{LK}_{i,j})$ is satisfied, the H-sensor confirms the validity of the L-sensor i; if not, H-sensor discards the response message. Hence, the H-sensor j can use this pairwise key to announce the message or new key K_l of the key-chain to the L-sensor i in the cluster.

(4) Then, the H-sensor j transmits the group key K_0 for two members in the cluster using the appropriate pairwise key, where K_0 is the first key in the key-chain. All subsequent messages transmitted within the cluster are encrypted by the K_0. The format of new key message is as follows:

$$\text{HID}_j \;||\; \text{Location of the L-sensor} \;||\; \{K_0\}_{\text{LK}_{i,j}}. \tag{4}$$

(5) After determining all the clustering nodes, the H-sensor j broadcasts the ID of members to all the nodes using K_0. If the H-sensor receives the response message from node u and node v simultaneously, the H-sensor judges whether node u and node v are neighbors based on the locations. However, this method does not always produce accurate results. If there is a barricade between node u and node v, it does not have an effect on the security. After judging whether the L-sensors are adjacent, the H-sensor sends all the L-sensor's IDs to the nodes. The format of neighbor message is as follows:

$$\text{HID}_j \;||\; \text{neighbor message} \;||\; \{\text{list of all neighboring nodes IDs}\}_{K_0}. \tag{5}$$

3.2. Normal Operations of HSNs. In the proposed system, the BS generates a key-chain for broadcasting and encrypting messages to the H-sensors. This process is very similar to what the H-sensor does for the L-sensor, as described in Sections 3.1.1 and 3.1.2. To simplify the description of the system structure, this paper omits the details of these procedures. This paper assumes that the BS has generated a key-chain and used the key, say K_{BS}, and pairwise key HK_j (the same as cluster key) for all the H-sensors.

This section discusses two different scenarios for the normal operations of the HSNs. Scenario 1 is that the BS broadcasts a message to all the H-sensors to gather the data from all the L-sensors. Scenario 2 is that the BS asks the H-sensor j to request the data from the specific L-sensor i.

Scenario 1. Figure 3 shows that the BS broadcasts the message using key-chain key K_{BS} to all the H-sensors for requesting to gather the data from the HSNs. The H-sensor then uses the cluster key K_i to communicate with the L-sensors.

Scenario 2. Figure 4 shows that the BS sends the demand using pairwise key HK_j to H-sensor j to request the data from L-sensor i in the HSNs. H-sensor j uses the pairwise key $LK_{i,j}$ to communicate with the L-sensor i.

3.3. Adaptability of the Proposed Method. This section discusses the adaptability of the proposed method, including key revocation, addition of a new node, and the generation of a new key-chain.

3.3.1. Key Revocation. In HSNs, if the BS discovers a compromised node or adversary (assuming in this study that the BS has an intrusion detection system mechanism inside), the BS broadcasts the following message to all the H-sensors:

$$\text{Malicious node message} \;||\; \{\text{MAC}(\text{LID}_x) \;||\; \text{location of the node } x \;||\; \text{node's } \text{LID}_x\}_{K_{\text{BS}}}. \tag{6}$$

Assuming that node u is a compromised node, H-sensor j will transmit the revocation message to remove the ID of node u from the other members in the cluster. H-sensor j then uses the pairwise key to encrypt the new key for L-sensor. This method ensures that the compromised node does not receive the new key and the old key is revoked. The

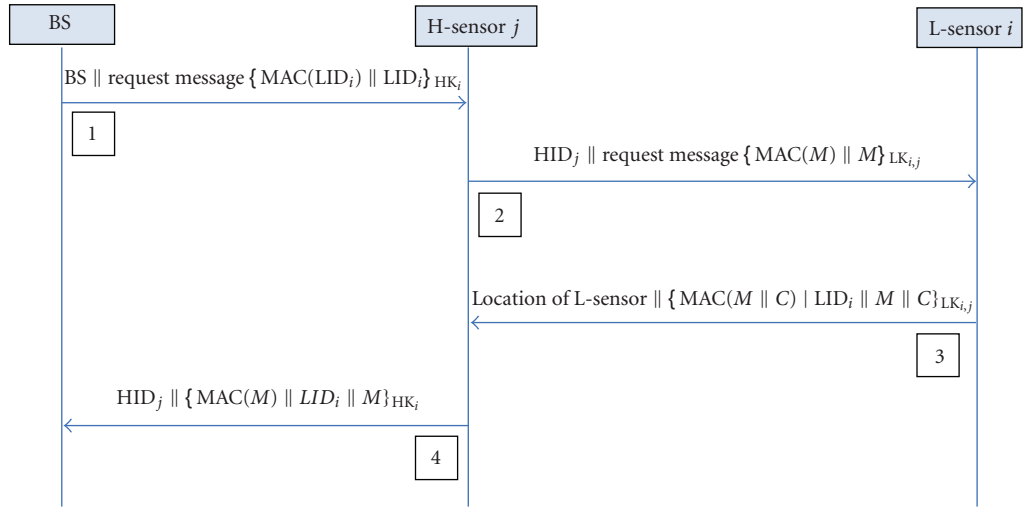


Figure 4: BS requires the data from the specific L-sensor.

format of key revocation message that H-sensor sends to the L-sensor x is as follows:

$$\mathrm{HID}_j \,||\, \text{revocation message} \,||\, \{\text{remove } u\text{'s ID form IDs list} \,||\, K_{i+1} \,||\, R_{i+1}\}_{LK_{x,j}}. \tag{7}$$

The L-sensor x confirms the K_i using K_{i+1} and R_{i+1}. If $K_i = H(K_{i+1} \oplus R_{i+1})$ is satisfied, they use K_{i+1} to send messages to each other. Otherwise, L-sensor x discards the message.

3.3.2. Addition of a New Node. The newly deployed node needs to establish pairwise key with its own H-sensor. Before adding new node into an environment, this new node should be ensured that it is not a comprised node and the hash function $H(\cdot)$ and the temporary session key K'_T are securely stored. After the deployment of a new L-sensor x, the BS actively delivers the following message about the addition of a new node to all H-sensors:

$$\text{new node message} \,||\, \{\mathrm{MAC}(\mathrm{LID}_x) \,||\, \text{new node}'\text{s } \mathrm{LID}_x, K'_T\}_{K_{\mathrm{BS}}}. \tag{8}$$

In this scheme, L-sensor x is deployed randomly in the environment. The L-sensor x will immediately broadcast a request message to all the neighboring H-sensors, where the message includes the L-sensor's ID LID_x encrypted by K'_T. If there are more than one H-sensor that received the request message from node x, then H-sensors will reply with a random number RN'_H to the node x by using K'_T with maximum power. The L-sensor x chooses an H-sensor j as its cluster head according to the distance and best signal strength of the message that replies to it. Hence, the node x and H-sensor j will generate the pairwise key $\mathrm{LK}_{x,\,j}$ by using the RN'_H, LID_x, and K'_T, as in Figure 5.

After generating the $\mathrm{LK}_{x,j}$, the H-sensor uses it to send the $R_m \,||\, R_{m-1} \ldots \,||\, R_1 \,||\, K_m \,||\, K_0$ in a message to L-sensor x, where R_m and K_m are the current random number and key in the key chain used by the H-sensor, respectively. The new L-sensor verifies if $K_0 = H(H(H(K_m \oplus R_m) \oplus R_{m-1}) \cdots \oplus R_1)$ is satisfied. If yes, then L-sensor x confirms the validity of the key K_1 to K_m and H-sensor j. Otherwise, the L-sensor x discards its message, and will select another H-sensor. Finally, L-sensor x then transmits the message to the H-sensor using K_m, and then H-sensor j broadcasts the neighbor message to all the members once again.

3.3.3. Generation of a New Key-Chain. When the last key K_S in the key-chain has been used in the cluster, as long as H-sensor still has sufficient power, it creates a new key-chain for the L-sensors in the cluster. H-sensor j uses the pairwise key to encrypt the new key for the L-sensors. The format of the message that the H-sensor sends to L-sensor x is as follows:

$$\mathrm{HID}_j \,||\, \text{Location of the L-sensor} \,||\, \{\mathrm{MAC}(K_0) \,||\, K_0\}_{\mathrm{LK}_{i,j}}. \tag{9}$$

4. Robustness to Attacks

A malicious node can be either an outside node that does not know the K_l in the key-chain or pairwise keys or a node that is captured by an adversary and becomes an internal compromised node.

This section classifies all potential attacks into five categories, such as guessing attacks, replay attacks, man-in-the-middle attacks, node capture attacks, and denial of service attack.

4.1. Guessing Attacks. Guessing attacks are a crucial concern in any security-based system. Assume that an adversary can obtain information or data related to the K_i in the HSNs. Based on this public information, it may be able to guess the K_i. However, the H-sensor will change the K_i to K_{i+1} at regular intervals. Further, each L-sensor node can use the pairwise key to encrypt messages to the H-sensor.

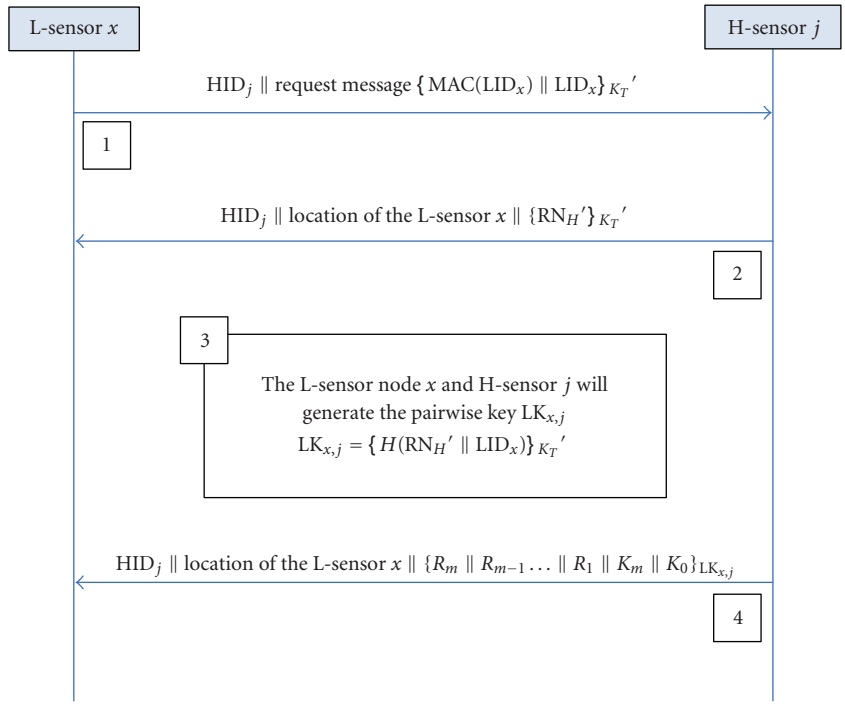


Figure 5: A pairwise key established between the node x and H-sensor j.

Therefore, the guessing attack does not have any effect in this environment.

4.2. Replay Attacks. The L-sensor i transmits the following message to the H-sensor j : HID_j || $\{\text{MAC}(M \,||\, C) \,||\, \text{LID}_i \,||\, M \,||\, C\}_{\text{LK}_{i,j}}$. The message includes H-sensor's ID, plain text M, MAC, L-sensor's ID, and a count C. When L-sensor i delivers the message it increases the C once. After the H-sensor j receives the message from the L-sensor i, it checks the value of C to determine if the node suffers from the replay attack.

4.3. Man-in-the-Middle Attacks. Man-in-the-middle attacks are a type of eavesdropping in which the adversary makes independent connections with the nodes and takes over the handling of messages between an L-sensor and the H-sensor. This attack fools sensors into thinking that they are communicating directly with each other over a private connection, when in fact all the details are controlled by the adversary. Based on the rules of the communication between nodes, the L-sensor and the H-sensor use a pairwise key or group key to securely and directly transmit messages to each other (as do the H-sensor and the base station). Therefore, if an adversary does not have the pairwise key or group key, it still cannot eavesdrop or modify the content of the message. Therefore, the man-in-the-middle attack does not have any effect on HSNs.

4.4. Node Capture Attacks. It is difficult to prevent this type of attacks if nodes are not tamper-proof and the environment is unattended. Hence, after all the L-sensors are deployed in the environment, the attacker might acquire some material of the K_T and $\text{LK}_{i,\,j}$ from the L-sensor i using node capture attack. However, the K_T is used twice in authentication phase and is discarded after the establishment of a pairwise key. In our scheme, each L-sensor has a different pairwise key in the cluster. Therefore, based on the property of pairwise keys, if the L-sensor i is captured by the adversary and it can gain the interior material of L-sensor i, it still cannot obtain the interior material of L-sensor x and cannot infect others.

4.5. Denial-of-Service Attacks. Denial-of-service attacks are common attacks in networks, where communication channel in HSNs is public. However, this type of attacks can be detected by enabling the network with an intrusion detection system. The proposed scheme provides protection against this attack. This is because it uses a one-way hash function and MAC in which the H-sensor sends message without expecting any acknowledgement. If the adversary prevents the message from reaching the nodes, neither the H-sensor nor the L-sensor will know about it.

5. System Analysis

This paper analyzed the proposed method from the following three issues: (1) the number of messages for grouping and establishing the pairwise key; (2) the key sizes; (3) the power consumption analysis. The H-sensors and L-sensors are randomly deployed in 500-square-meter wireless sensor

Table 2: Number of messages for compared methods.

			Pairwise key establishment	Key revocation
Kausar et al. [23]	H-sensor	transmitted	4	1
		received	2	0
	L-sensor	transmitted	2	0
		received	4	1
Our Scheme	H-sensor	transmitted	2	1
		received	1	0
	L-sensor	transmitted	1	0
		received	2	1

network. This HSN has two types of sensors: a few powerful H-sensor nodes and many L-sensor nodes. The ratio between these two types of sensors is 1 : 10. In our experiments, there are 25 H-sensor nodes and 250 L-sensor nodes. The H-sensor nodes have a key-chain length of 50 keys. The L-sensors are ordinary sensor nodes that are limited in terms of processing capability, power, and memory. They acquire data from the surrounding environment and forward it to the H-sensor nodes. The H-sensor nodes then transmit the data to the base station.

5.1. The Number of Messages between the H-Sensor and L-Sensor. This section compares the proposed scheme with other key distribution techniques. In the proposed scheme, each H-sensor establishes a pairwise key with its own L-sensor and three messages are exchanged: the H-sensor broadcasts two messages, and an L-sensor node sends one-response message. In updating the key, the H-sensor and L-sensor nodes only send one message, where the H-sensor node broadcasts the hello message, as Table 2 shows. Although Kausar et al. [23] has approximate number of messages that come to us in two phases, the proposed method would consume less energy for L-sensors in large HSNs.

5.2. The Key Sizes. This study compares the proposed scheme with the other three methods, which are q-composite keys [6], EPKEM [22], and the method of Kausar et al. [23]. These schemes have some properties similar to those of ours such as storing keys in sensor nodes before deployment and having pairwise keys. Cheng and Agrawal [22] and Kausar et al. [23] also compared their methods with q-composite keys [6] in their papers. Figure 6 shows a comparison chart on the number of keys for the proposed method and others. To maintain the probability of key connection, previous approaches [6, 22, 23] need more nodes in the environment, meaning that more keys are stored in the sensor nodes. In the proposed scheme, regardless of the number of L-sensor nodes, each L-sensor only stores three keys. This approach reduces memory space requirements and increases the efficiency of each sensor node.

In our environment, H-sensor node has an average of $14 + n$ keys, where 10 keys are pairwise keys of L-sensors,

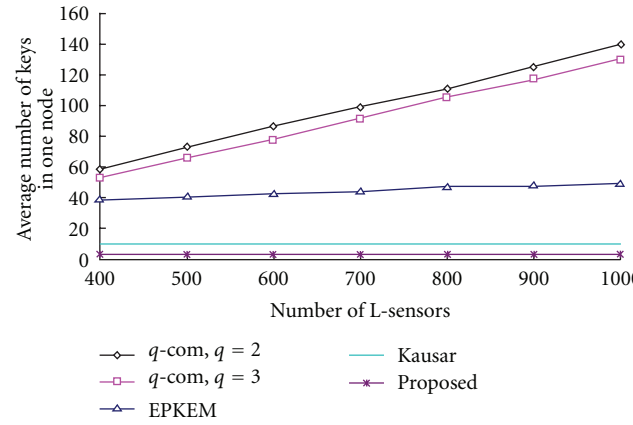


Figure 6: The comparisons of number of keys stored in the L-sensor.

Table 3: The number of keys and functions stored in each member of HSNs.

	Base station	H-sensor	L-sensor
Keys	P	64	3
Hash function	1	1	1

4 keys are HK, K_S, K_{BS}, and K_T, and n keys are the length of key-chain. Experimental results show that the length of key-chain in each H-sensor is 50 keys. Therefore, each H-sensor must store 64 keys in the HSNs. The L-sensor only stores 3 keys, which are K_I, K_T, and $LK_{i,j}$. Table 3 shows the number of keys for each member.

5.3. Power Consumption Analysis. In this section, we will run a simulation to show the power consumption of the proposed scheme. The number of survival nodes as time goes by is used as a metric for power consumption. For each sensor node, the costs of the energy consumption are primarily in data transmission and receiving. As in the work of Zhang et al. [27], a mote of Crossbow MICA2DOT with a Chipcon CC1000 radio device consumes 28.6 uJ and 59.2 uJ for receiving and transmitting one byte of packet, respectively. ZigBee specifies a maximum packet length of 128 byte in which 100 byte for the payload, 20 byte for the header, and 8 byte for preamble; the preamble consists of source, destination, packet ID, and a control byte. In our scheme, we assume that a packet consists of 16-byte MAC (the size of hash, 128 bit), 16-byte payload, 20-byte header, and 10-byte preamble. The total length of packet is 62 bytes. Each L-sensor node is assigned an initial energy of 1 J, and the power consumption for receiving and transmitting one byte of packet is assumed to be 28.6 uJ and 59.2 uJ, respectively. Figure 7 shows the number of survival nodes over time.

In the simulation, the proposed scheme is compared with the normal HSN without key management rather than the other key management schemes. This is due to the lack of power consumption evaluation in other HSN key management schemes. The power consumption of the proposed scheme was evaluated in terms of the number of

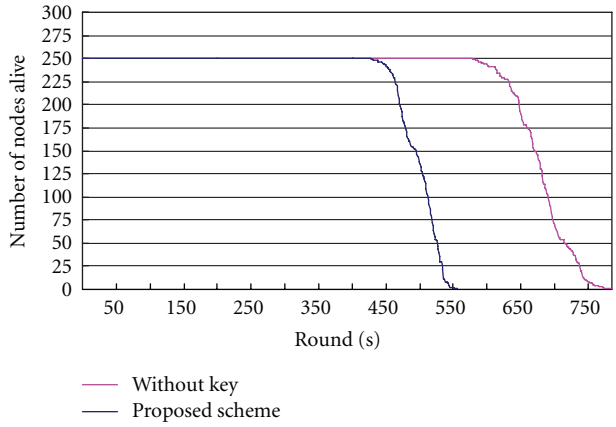


Figure 7: Number of survival nodes over rounds.

survival nodes over rounds. Each round in the simulation is defined as the completion of one of the following three tasks: (1) H-sensor requests and receives the data from all the L-sensors in the cluster; (2) H-sensors requests and receives the data from a specific L-sensor in the cluster; (3) key revocation. For normal HSN without key management, only the first two operations are possible, and the packet length of 46 bytes is also assumed.

The experimental results are shown in Figure 7. The first sensor node that ran out of power occurred at the 428th round in the proposed scheme in contrast to the 579th round for the normal HSN without key management. The whole network died at about the 557th round and the 786th round for the proposed scheme and the normal HSN without key management, respectively. As a result, the proposed method incurred about 29% overhead due to the inclusion of key management scheme. But considering the benefits of the proposed scheme, which include protections against the guessing attacks, replay attacks, and man-in-the-middle attacks as discussed in Section 4, we think the overhead is acceptable and the results could be a starting point for evaluating power consumption on sensor networks with key management.

6. Conclusion

This study proposes a new key management scheme that is suitable for HSNs. By clustering all the sensor nodes in the environment, cluster heads can generate their own key-chain. The sensor nodes and their cluster heads can jointly establish pairwise keys. Pairwise keys ensure transmission secrecy for each message, protecting data integrity and determining if the sensor nodes are malicious. The key-chain consists of continuous keys, and each key is dependent. This makes it possible for the sensor node to confirm the validity of each key. Sensor nodes or cluster heads through the characteristic of key-chain, when the cluster heads change the key, and then sensor nodes can confirm the identity of the cluster head and the validity of new key. In our scheme, the key is calculated by hash function. The hash function makes it possible to compress data into a fixed length and avoid data collision. Sensor nodes only need to store a few keys and a hash function at a time, reducing the memory requirements of sensor nodes and ensuring key security.

Acknowledgment

This work was supported partially by National Science Council, Taiwan under Grant NSC 98-2221-E-005-083.

References

[1] L. Kejie, Q. Yi, and H. Jiankun, "A framework for distributed key management schemes in heterogeneous wireless sensor networks," in *Proceedings of the 25th IEEE International Performance, Computing, and Communications Conference (IPCCC '06)*, pp. 513–519, April 2006.

[2] D. Liu, P. Ning, and L. I. Rongfang, "Establishing pairwise keys in distributed sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 1, pp. 41–77, 2005.

[3] D. E. Knuth, *The Art of Computer Programming*, Addison-Wesley, Reading, Mass, USA, 2nd edition, 1981.

[4] J. Nechvatal, "Public key cryptography," in *Contemporary Cryptology: The Science of Information Integrity*, G. Simmons, Ed., IEEE Press, Piscataway, NJ, USA, 1992.

[5] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02)*, pp. 41–47, November 2002.

[6] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the IEEE Symposium on Security And Privacy*, pp. 197–213, May 2003.

[7] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 24–34, 2007.

[8] D. Liu and P. Ning, "Location-based pairwise key establishments for static sensor networks," in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 72–82, October 2003.

[9] G. Li, J. He, and Y. Fu, "Key management in sensor networks," in *Proceedings of the International Conference on Wireless Algorithms, Systems and Applications*, pp. 457–466, August 2006.

[10] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Securing sensor networks with location-based keys," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '05)*, pp. 1909–1914, March 2005.

[11] A. Wadaa, S. Olariu, L. Wilson, and M. Eltoweissy, "Scalable cryptographic key management in wireless sensor networks," in *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops*, pp. 796–802, March 2004.

[12] R. Blom, "Non-public key distribution," in *Proceedings of the International Cryptology Conference on Advances in Cryptology (CRYPTO '98)*, pp. 231–236, 1998.

[13] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '93)*, pp. 471–486, 1993.

[14] D. Liu, P. Ning, and W. Du, "Group-based key pre-distribution in wireless sensor networks," in *Proceedings of the 4th ACM*

Workshop on Wireless Security (WiSe '05), pp. 11–20, September 2005.

[15] M. A. Moharrum and M. Eltoweissy, "A study of static versus dynamic keying schemes in sensor networks," in *Proceedings of the 2nd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN '05)*, pp. 122–129, October 2005.

[16] M. Eltoweissy, M. Moharrum, and R. Mukkamala, "Dynamic key management in sensor networks," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 122–130, 2006.

[17] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: security protocols for sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 189–199, July 2001.

[18] M. F. Younis, K. Ghumman, and M. Eltoweissy, "Location-aware combinatorial key management scheme for clustered sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 8, pp. 865–882, 2006.

[19] G. Jolly, M.C. Kuscu, P. Kokate, and M. Younis, "A low-energy key management protocol for wireless sensor networks," in *Proceedings of the 8th IEEE international Symposium on Computers and Communications*, pp. 335–340, 2003.

[20] H. Chan and A. Perrig, "Pike: peer intermediaries for key establishment in sensor networks," in *Proceedings of the 24th annual joint conference of the IEEE computer and communications societies (INFOCOM '05)*, pp. 524–535, 2005.

[21] L. Guorui, H. Jingsha, and F. Yingfang, "A group-based dynamic key management scheme in wireless sensor networks," in *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops/Symposia, (AINAW '07)*, pp. 127–132, May 2007.

[22] Y. Cheng and D. P. Agrawal, "Efficient pairwise key establishment and management in static wireless sensor networks," in *Proceedings of the 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS '05)*, pp. 544–550, November 2005.

[23] F. Kausar, S. Hussain, L. T. Yang, and A. Masood, "Scalable and efficient key management for heterogeneous sensor networks," *Journal of Supercomputing*, vol. 45, no. 1, pp. 44–65, 2008.

[24] G. Gupta and M. Younis, "Load-balanced clustering of wireless sensor networks," in *Proceedings of the International Conference on Communications (ICC '03)*, vol. 3, pp. 1848–1852, May 2003.

[25] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS '00)*, pp. 1–10, January 2000.

[26] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the 7th ACM/IEEE Annual International Conference on Mobile Computing and Networking (MobiCom '01)*, pp. 166–179, July 2001.

[27] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 247–260, 2006.