A Multiple-Key Management Scheme in Wireless Sensor Networks

William Chu, Jung-Chun Liu, Yi-Li Huang, Fang-Yie Leu, Ilsun You, Feng-Ching Chiang, Chao-Tung Yang

A Multiple-Key Management Scheme in Wireless Sensor Networks

Jung-Chun Liu, Yi-Li Huang, Fang-Yie Leu*, Ilsun You[#1], Feng-Ching Chiang, Chao-Tung Yang, William Cheng-Chung Chu

Department of Computer Science, TungHai University, Taiwan
{jcliu, yifung, leufy, g01350011, ctyang, cchu}@thu.edu.tw
1: School of Information Science, Korean Bible University, South Korea
#: ilsunu@gmail.com

Abstract. In a wireless sensor network (WSN), in order to provide a secure communication environment for all the sensor nodes, we often securely authenticate network nodes and protect all the messages delivered among them. When a sensor node (or simply a node or a sensor) newly joins a WSN, it is required for the Key Distribution Server (KDS) to distribute those generated security keys to this node and all the existing nodes before they can securely communicate with each other. But due to the wireless nature, when a node broadcasts a message M, all its neighbors can receive M. To securely protect this message, a security mechanism is required. Therefore, in this paper we propose a Multiple-key Management Scheme (MMaS for short), in which a sensor N receives two sets of keys from the KDS when the system starts up. The first set, named communication keys, is used by N to securely communicate with its neighbor sensors; the other, called the individual key, is employed to encrypt messages delivered between N and the KDS. When N would like to communicate with another node, e.g., M, they exchange their IDs with each other so as to correctly identify their common keys (CKs), which are used to individually generate a shared key (SK) on both sides for encrypting/decrypting messages transmitted between them. When N leaves the underlying network, the CKs currently related to N can be reused by a newly joining sensor, e.g., M. However, when M joins the network, if no such used ID is available, M will be given a new ID and CKs by the KDS. The KDS will encrypt the CKs, that will be used by an existing node H to communicate with M, with the individual key of H so that only H rather than M can correctly decrypt the CKs, with which to securely communicate with M. The security analysis shows that the proposed system is secure.

***Keywords*:** Multi-key management scheme, Key distribution server, Newly joining node, Shared key, Wireless sensor network, An incrementally constructed system

1 Introduction

Wireless sensor networks (WSNs) are envisioned to be widely applied to commercial and military applications [1], such as target tracking, health-care [2], environmental monitoring [3] and homeland security. However, some applications require certain security mechanisms to verify the source of a message and protect the integrity of

transmitted data from being maliciously modified. In order to securely authenticate a network entity and deliver messages, a secure communication environment is required. To build a secure sensor network, Wuu et al. [4] proposed a Quorum-based Key Management Scheme. But it has a problem in sensor node addition since the number of sensor nodes (or simply sensors or nodes) must be odd. So each time at least two sensor nodes must be added. Furthermore, when two nodes are newly added to a sensor network, the shared keys (SKs) of some existing sensor nodes are changed**.** We will show this later. This may crash the normal operation of the whole system.

Generally, an asymmetric cryptographic technique generates many large numbers to encrypt keys and delivered messages. But this is infeasible for WSNs, since sensor nodes are often powered by battery and provided with very limited processing capability. Therefore, to achieve high security and support sensor node addition functionality, by which extra nodes can be easily added to a sensor network, in this study we propose a symmetric cryptographic technique, named the Multi-key Management Scheme (MMaS), based on a $n \times n$ key matrix K, different parts of which are distributed among sensors. Further, due to the fast advancement of hardware technology, memory equipped in sensors is cheaper than before and the size of a WSN grows rapidly in recent years. This further makes MMaS feasible in practical applications.

2 Related Works

Various key pre-distribution schemes used to establish secure channels for wireless sensors have been proposed in literature [4][5]. The key pre-distribution scheme of Cheng et al. [5] used a $\sqrt{n} \times \sqrt{n}$ matrix as a key matrix to assign keys to sensors, where n is the total number of sensors in the system. Fig. 1 illustrates an example of a key matrix in which the intended network size is n. The scheme has two phases: the key pre-distribution phase and pair-wise key setup phase. At first, the KDS randomly selects n keys from its key pool, in which there are more than 2^{20} distinct keys, and uses the n keys to construct an m x m key matrix K, where $m = \sqrt{n}$. The KDS assigns an element of this matrix, e.g., $K_{i,j}$, as the sensor's ID, and the other entries on the ith row and jth column as the sensor's keys, implying that the matrix is indexed by the IDs of these involved sensors. It also means that this scheme provides the largest maximum supported network size since each element of the matrix represents one sensor node. When a sensor M would like to communicate with another sensor, e.g., N, it identifies the CK indexed by M and N and uses it to encrypt messages.

$K_{1,1}$	$K_{1,2}$	$K_{1,3}$	$K_{1,4}$	$K_{1,5}$
$K_{2,1}$	$K_{2,2}$	$K_{2,3}$	$K_{2,4}$	$K_{2,5}$
$K_{3,1}$	$K_{3,2}$	$K_{3,3}$	$K_{3,4}$	$K_{3,5}$
$K_{4,1}$	$K_{4,2}$	$K_{4,3}$	$K_{4,4}$	$K_{4,5}$
$K_{5,1}$	$K_{5,2}$	$K_{5,3}$	$K_{5,4}$	$K_{5,5}$

Fig. 1 An example of a key matrix

As stated above, Wuu et al. [4] proposed a Quorum-based Key Management Scheme, in which the KDS as shown in Fig. 2 generates a $\lfloor n/2 \rfloor \times n$ key matrix K and

establishes a quorum system S based on K, where each sensor, e.g., j, has the entire column j of matrix K and $\lfloor n/2 \rfloor$ other elements, each belonging to one of the $\lfloor n/2 \rfloor$ columns after column j, meaning that each sensor has $n-1$ elements, i.e., $K_{i,j}, 1 \leq i \leq \lfloor n/2 \rfloor$, and $K_{i,j+i \bmod n}, 1 \leq i \leq \lfloor n/2 \rfloor$, $1 \leq j \leq n$. As shown in Fig. 3, after the deployment of sensors, two arbitrary sensors, e.g., A and B, both can individually identify the CKs assigned to them so that they can mutually authenticate and communicate with each other. In this scheme, node addition is feasible only when some existing IDs that are not in use are available. Also, when two nodes A and B newly join the WSN, as shown in Fig. 4, the SKs of some nodes will be changed. For example, originally the CK of nodes 1 and 5 was $K_{1,1}$. But after sensors A and B join the network, the CK of nodes 1 and 5 becomes $K_{4,5}$. After that, the normal operations of the system will be destroyed.

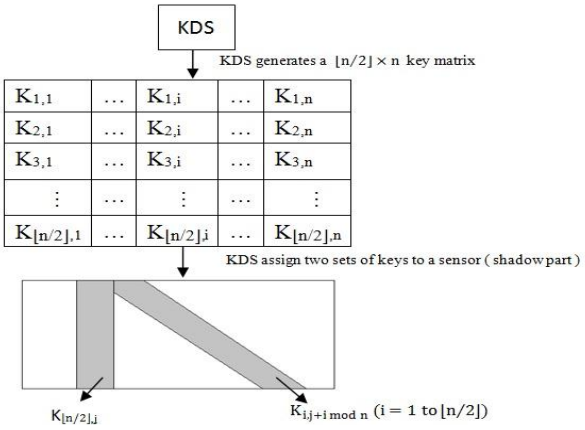

Fig. 2 KDS assigns each sensor node two sets of keys (the shadowed parts)

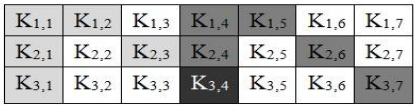

Fig. 3 Sensors A and B derive a common key

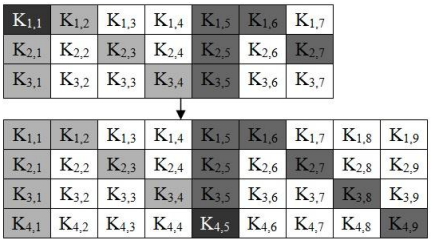

Fig. 4 New sensor nodes A (node 8) and B (node 9) join the network

3 The Proposed Scheme

The MMaS consists of four working phases: the key pre-distribution, shared key establishment, key refreshment, and data transmission phases. In the key pre-distribution phase, the KDS generates a $n \times n$ key matrix, in which the keys are random numbers. After that, the KDS assigns these keys to sensors during the deployment of sensor nodes. Before communicating with each other, each pair of sensors needs to identify the other's CKs and then generates the SK in the shared key establishment phase. When the sensor, e.g., M, joins the network, the KDS broadcasts the ID, i.e., M, and the related CKs, and the system enters its key refreshment phase, in which the receiving sensor accordingly updates its key information. In the data transmission phase, sensors transmit data to their neighbors, and check to see whether a received message is sent by a legal sensor or not.

3.1 Key pre-distribution phase

Each sensor i has two sets of keys. The first set, named communication keys (row i and column i together are called key-cross i), is used to perform one-to-one communication between two sensors by using the computed SK. The other one, called individual key, e.g., $k_{i,i}$, is the key employed by sensor i to communicate with the KDS where $K_{i,i}$ is the i^{th} element along the diagonal of the key matrix. The steps of the key pre-distribution phase are as follows:

Step 1: the KDS generates n^2 random numbers to establish the $n \times n$ key matrix K.

Step 2: the KDS assigns an ID, e.g., i, which is also the index of key $K_{i,i}$, $1 \leq i \leq n$, and two common keys $K_{i,j}$ and $K_{j,i}$ (see Fig. 5), $1 \leq i, j \leq n$, to a sensor. After that, sensors i and j can securely communicate with each other by encrypting those messages exchanged between them with the SK derived from $K_{i,j}$ and $K_{j,i}$. As stated above, the individual key of sensor i, i.e., $K_{i,i}$, $0 < i < n$, is used to encrypt messages delivered between sensor i and the KDS.

$K_{1,1}$	$K_{1,2}$	$K_{1,3}$	$K_{1,4}$	$K_{1,5}$
$K_{2,1}$	$K_{2,2}$	$K_{2,3}$	$K_{2,4}$	$K_{2,5}$
$K_{3,1}$	$K_{3,2}$	$K_{3,3}$	$K_{3,4}$	$K_{3,5}$
$K_{4,1}$	$K_{4,2}$	$K_{4,3}$	$K_{4,4}$	$K_{4,5}$
$K_{5,1}$	$K_{5,2}$	$K_{5,3}$	$K_{5,4}$	$K_{5,5}$

Fig. 5 The KDS generates the $n \times n$ key matrix K, in which $[K_{1,i}, K_{2,i}, \cdots, K_{i,i}, \cdots, K_{n,i}]$ and $[K_{i,1}, K_{i,2}, \cdots, K_{i,i}, \cdots, K_{i,n}]$ are assigned to sensor i.

3.2 Dynamic shared key establishing phase

After the deployment of sensors, when sensor A would like to communicate with sensor B, it sends his own ID, i.e., A, to B. With the two IDs, both side can individually identify the CKs $K_{A,B}$ and $K_{B,A}$, $1 \leq A, B \leq n$.

Before authenticating node B (node A), node A (node B) generates an authentication message $Auth_A$ ($Auth_B$) which contains the result of a hash function with the

concatenation of the CK, i.e., $K_{A,B}$ ($K_{B,A}$), and a random number randA (randB) as its parameter where,

$$\text{AuthA} = \text{H}(K_{A,B}||\text{randA}) \qquad (1)$$

and

$$\text{AuthB} = \text{H}(K_{B,A}||\text{randB}) \qquad (2)$$

After that, A (B) delivers the message it generates, i.e., AuthA (AuthB), and randA (randB) to B (A). On receiving AuthB and randB, node A retrieves $K_{B,A}$ from its own common keys and invokes Eq. (2) with the concatenation of the retrieved $K_{B,A}$ and the received randB as its parameter to calculate AuthB, denoted by AuthB_c, and then checks to see whether the received AuthB, denoted by AuthB_r, is equal to AuthB_c or not. If yes, meaning B is a legal one, it retrieves the common keys $K_{A,B}$ and $K_{B,A}$ from its own key array and generates the SK, where

$$\text{SK} = K_{A,B} \oplus K_{B,A} \qquad (3)$$

In fact, node B does similarly. But the invoked Equation is Eq. (1) and the checked equation is $\text{AuthA}_c = \text{AuthA}_r$. If they are equal, B generates SK also by invoking Eq. (3).

3.3 Key refreshment phase

There are two cases which call for key refreshment. One is when a sensor leaves or joins the network; the other is when CKs of two sensors have been used in communication over a threshold of number.

When a sensor N leaves the network, the KDS broadcasts N to the remaining sensors. On receiving the information, a sensor will no longer communicate with N. In fact, this ID is now available and can be reused. When a sensor, e.g., M, newly joins the network, it faces two situations: with or without an available used ID in the underlying network.

1. If the situation "with an available used ID" occurs, the KDS assigns the ID to M and generates CKs, denoted by $K_{M,N}$ and $K_{N,M}$, used by M and an existing sensor N to communicate with each other, $1 \leq \text{M}, \text{N} \leq \text{n}, \text{M} \neq \text{N}$.
2. If the situation "without an available used ID" occurs, the KDS will generate a new ID and the corresponding $K_{M,N}$ and $K_{N,M}$. After that, the KDS broadcasts a message containing M's ID and those generated $K_{M,N}$ and $K_{N,M}$, $1 \leq \text{N} \leq \text{n}$, to all sensors. In this message, the CKs deliverd to sensor N is encrypted by $K_{N,N}$ so that only sensor N can decrypt the CKs sent to it. In our scheme, the addition of M (maybe $\text{M} = \text{n} + 1$ or $1 \leq \text{M} \leq \text{n}$) does not change those SKs currently used by existing sensors.

The format of this broadcasted message is shown in Fig. 6, in which a sensor ID, e.g., N, is followed by the common keys, i.e., $K_{M,N}$ and $K_{N,M}$, needed to be updated by sensor N. Upon receiving this message, sensor N sequentially searches the ID field. When ID

= N as the head field of sensor N is identified, sensor N decrypts the common keys conveyed in fields following the head field, and accordingly updates its key matrix.

For example, if ID = 1, the following two fields are $K_{1,1} \oplus rand_{n+1,1}$ and $K_{1,1} \oplus rand_{1,n+1}$ where $rand_{n+1,1}$ is $K_{n+1,1}$ and $rand_{1,n+1}$ is $K_{1,n+1}$. Only the legal KDS has the right individual key to encrypt the two fields, and only the legal sensor, i.e., sensor 1, which has $K_{1,1}$, is able to decrypt the two fields. Moreover, after obtaining $rand_{n+1,1}$ and $rand_{1,n+1}$, sensor 1 compares the $(rand_{n+1,1} \oplus rand_{1,n+1})$ generated by itself with the fourth field from the head field to see whether they are equal or not. If yes, the message is authenticated.

ID=1	$K_{1,1} \oplus rand_{n+1,1,}$	$K_{1,1} \oplus rand_{1,n+1}$	$rand_{n+1,1} \oplus rand_{1,n+1}$
ID=2	$K_{2,2} \oplus rand_{n+1,2}$	$K_{2,2} \oplus rand_{2,n+1}$	$rand_{n+1,2} \oplus rand_{2,n+1}$
	...		
ID=n	$K_{n,n} \oplus rand_{n+1,n}$	$K_{n,n} \oplus rand_{n,n+1}$	$rand_{n+1,n} \oplus rand_{n,n+1}$

Fig. 6 Format of the message containing sensor IDs, e.g., i, and the common Keys, $rand_{n+1,1,}$ and $rand_{1,n+1}$ needed to be updated by sensor i

When the number of communication between sensors A and B is over a predefined threshold, due to security consideration, A and B need to refresh the CK in their upper triangle of K, i.e., $K_{A,B}$ (without changing $K_{B,A}$), by executing the following key refreshment steps under the assumption that $A < B$.

Step 1: Input the CK, i.e., $K_{A,B}$, which is in the upper triangle of the key matrix K, to a predefined hash function to produce a new key, e.g., $K'_{A,B}$, where

$$K'_{A,B} = H(K_{A,B}) \quad (4)$$

Step 2: Compute the new shared key SK' where

$$SK' = K'_{A,B} \oplus K_{B,A} \quad (5)$$

Step 3: Store the SK' as the new shared key in its local variable.

3.4 Data transmission phase

After completing the authentication, two sensors can communicate with each other by sending a message, the format of which is shown in Fig. 7.

ID	$msg \oplus SK$	HMAC

Fig. 7 The format of a data message

When receiving this message, a sensor computes the hash value $HMAC = H(ID||msg \oplus SK, SK)$ and checks to see whether the value is equal to the one conveyed in the received message or not to ensure data integrity of the message where msg is the data that needs to be sent. Since only a legal sensor has the right SK to produce the correct hash value, if the two values are equal, the message is authenticated, meaning that the sensor sending this message is a legal one.

4 Security Analysis

By using shared keys and the unique individual key, $K_{i,i}$, a sensor node and the KDS, or any two sensor nodes can mutually authenticate each other so as to defend attacks launched by hackers. In this section, we analyze three common attacks, including the eavesdropping, forgery KDS, and forgery sensor node attacks, and show that the MMaS can effectively defend these attacks.

4.1 Eavesdropping attack

Due to the wireless nature, messages sent by sensor nodes and the KDS can be accessed by a sensor located within the communication area of the sender. As described above, since illegal users cannot decrypt messages protected by the multiple key, i.e., SK and $K_{i,i}$, the messages delivered in the data transmission phase is secure. So the eavesdropping attack does not work.

4.2 Forgery KDS attack

A forgery KDS may send fake messages intending to cheat sensors that some sensor nodes leave or newly join the network. This kind of attack can be prevented by the unique individual key, $K_{i,i}$, which is only known to the individual sensor and the KDS, and is used to encrypt messages and authenticate the integrity of a message delivered between the node and KDS. Only the legal KDS has the right $K_{i,i}$ and only the corresponding sensors can correctly use it to decrypt the messages issued by the KDS, meaning that the MMaS can effectively defend the forgery KDS attack.

4.3 Forgery sensor node attack

If a hacker, e.g., B', disguising itself as the legal sensor B, sends data messages more than the threshold number in order to falsely trigger CK refreshment between A and B' so as to make the CKs between A and B inconsistent. He/she will be defeated by MMaS, since sensors need to authenticate each other before communication. But the faked node does not have the right CKs. The authentication will fail. Thus B' is incapable of identifying the right SK for further interaction with sensor A. By this SK authentication mechanism, legal sensors will not respond to the faked one's CK refreshment attempt used to falsely alter the keys, and the faked one cannot decrypt messages issued by a legal one because it does not own the right SK.

5 Conclusions and Future Studies

In this paper, we design and analyze a multiple key management scheme, with which to securely protect wireless sensor networks. To increase the resiliency of sensor networks, our scheme supports an efficient sensor-node-addition mechanism to deal with the dilemma in which when a sensor network has no available used IDs, adding extra sensor nodes will change the SKs used by other nodes and may aggravate or even crash the whole system. We also analyze and show that the proposed system can effectively defend three common attacks. The system enhances the security and resiliency of the sensor networks without conducting tremendous amount of computation and complicated asymmetric cryptographic techniques.

In the future, we would like to improve reliability and derive working models for the proposed system. To further enhance performance and reduce the size of a delivered message, we plan to devise an authentication function to substitute for the random number keys and authentication messages. In other words, we only need to invoke a function instead of issuing n authentication messages to authenticate messages. These constitute our future studies.

Acknowledgment
The work was partially supported by TungHai University under the project GREENs and the National Science Council, Taiwan under Grants NSC 101-2221-E-029-003-MY3, and NSC 100-2221-E-029-018.

6 Reference

[1] Durisic, M.P., Tafa, Z., Dimic, G. and Milutinovic, V., "A Survey of Military Applications of Wireless Sensor Networks," Mediterranean Conference on Embedded Computing, pp. 196-199, June 2012.
[2] Chen Y.M, Shen W., Huo H.W. and Xu Y.Z., "A Smart Gateway for Health Care System Using Wireless Sensor Network," International Conference on Sensor Technologies and Applications, pp. 545-550, July 2010.
[3] Kong Y.F. and Jiang P., "Development of Data Video Base Station in Water Environment Monitoring Oriented Wireless Sensor Networks," International Conference on Embedded Software and Systems Symposia, pp. 281-286, July 2008.
[4] Wuu L.C., Hung C.H. and Chang C.M., "Quorum-based Key Management Scheme in Wireless Sensor Networks," International Conference on Ubiquitous Information Management and Communication, no. 15, 2012.
[5] Cheng, Y. and Agrawal, D.P., "Efficient Pairwise Key Establishment and Management in Static Wireless Sensor Networks," IEEE International Conference on Mobile Ad hoc and Sensor Systems, pp. 544-550, 2005.