A SQUARE KEY MATRIX MANAGEMENT SCHEME IN WIRELESS SENSOR NETWORKS

Jung-Chun Liu, Yi-Li Huang, Fang-Yie Leu*, Feng-Ching Chiang, Chao-Tung Yang, William Cheng-Chung Chu

Department of Computer Science, TungHai University, Taiwan
e-mail: {`jcliu,yifung,leufy,g0135011,ctyang,cchu`}`@thu.edu.tw`

Abstract. In this paper we propose a symmetric cryptographic approach named the Square Key Matrix Management Scheme (SKMaS for short) in which a sensor node, named the Key Distribution Server (KDS for short) is responsible for the security of key management. When the system starts up, the KDS sends its individual key and two sets of keys to sensor nodes. With the IDs, any two valid sensor nodes, e.g., i and j can individually identify the corresponding communication keys (CKs) to derive a dynamic shared key (DSK) for encrypting/decrypting messages transmitted between them. When i leaves the underlying network, the CKs and the individually keys currently utilized by i can be reused by a newly joining sensor, e.g., h. However, when h joins the network, if no such previously-used IDs are available, h will be given a new ID, CKs and the individually key by the KDS. The KDS encrypts the CKs, with which an existing node q can communicate with h, with individual key so that only q rather than h can correctly decrypt the CKs. The lemmas and security analyses provided in this paper prove that that the proposed system can protect at least three common attacks.

Keywords: Square-Key Matrix Management Scheme, Key distribution server, Newly joining node, Shared key, Wireless sensor network.

1 INTRODUCTION

Wireless sensor networks (WSNs) are envisioned to be widely applied to commercial and military applications [1, 2, 3, 4], such as target tracking [5], health-care [6, 7], environment monitoring [8, 9] and homeland security [10]. However, some WSN

applications require certain security mechanisms [11] 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 [12, 13] is required.

To build a secure WSN, Wuu et al. [14] proposed a Quorum-based Key Management Scheme. But this scheme has a problem in sensor node addition since the number of sensor nodes (or simply sensors or nodes in the following) 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. This may crash the normal operation of the whole system. We will show this later.

Generally, an asymmetric cryptographic technique [15] 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 [16, 17, 18, 19, 20]. Therefore, to achieve a high security level 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 approach, named the Square Key Matrix Management Scheme (SKMaS), in which a sensor node named the Key Distribution Server (KDS for short) which is the sensor with ID=1 is responsible for the security of key management. When the system starts up, the KDS delivers its individual key $K_{1,1}$, a control key $K_{0,0}$, and two sets of communication keys (CKs) to sensor node i, $2 \le i \le n$, where n is the number of nodes currently in the WSN. The first set of CKs is used by i to securely communicate with its neighbor sensor $j, 2 \le i, j \le n, i \ne j$; the other, also called i's individual key, is employed to encrypt messages delivered between i and the KDS.

When i would like to communicate with node j, the two nodes exchange their IDs with each other. With the IDs, i and j can individually identify the corresponding CKs with which to derive a dynamic shared key (DSK for short) for encrypting/decrypting messages transmitted between them. When i leaves the underlying network, the CKs and the individually keys currently used by i can be reused by a newly joining sensor, e.g., h. However, when h joins the network, if no such previously-used IDs are available, h will be given a new ID (e.g., $n + 1$), CKs, the control key $K_{0,0}$ and the individually key (e.g., $K_{n+1,n+1}$) by the KDS. The CKs, by which an existing node q can communicate with h, are encrypted by using the individual key $K_{q,q}$ by the KDS so that only q rather than h can correctly decrypt the CKs, $2 \le q \le n$, based on the $n \times n$ key matrix created by the KDS. Different parts of the matrix are distributed to different sensors. Furthermore, 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. The memory size of a sensor no longer constitutes a problem. This further makes SKMaS 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 [14, 16]. The key pre-distribution scheme proposed by Cheng et al. [21] introduced a $\sqrt{n} \times \sqrt{n}$ matrix as a key matrix, in which different parts of keys are assigned to different sensors where n is the total number of sensors in the system. 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 more than 2^{20} distinct keys have been collected. The KDS uses these keys to construct an $m \times m$ key matrix K, where $m = \sqrt{n}$. The KDS assigns an element of this matrix, e.g., $K_{i,j}$, as a sensor's ID and the other entries in the ith row and j^{th} column as the sensor's keys, to this sensor, implying that the matrix is indexed by the IDs of the 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 i would like to communicate with another sensor, e.g., j, it identifies the common keys indexed by i and j and uses them to encrypt those messages delivered between them [29].

As stated above, Wuu et al. [14] proposed a Quorum-based Key Management Scheme, in which the KDS as shown in Fig. 1a generates a $\lfloor n/2 \rfloor \times n$ key matrix K and establishes a quorum system based on K. Each sensor, e.g., j, has the entire column j of matrix K and $\lfloor n/2 \rfloor$ other elements. Each belongs to one of the $\lfloor n/2 \rfloor$ columns after column j, meaning that each sensor has $n - 1$ elements, i.e., $K_{i,j}$ and $K_{i,j+i \bmod n}$, $1 \le i \le \lfloor n/2 \rfloor$, $1 \le j \le n$. As shown in Fig. 1b, after the deployment of sensors, two arbitrary sensors, e.g., A and B, can individually identify the common keys assigned to them so that they can mutually authenticate and securely communicate with each other. In this scheme, node addition is feasible only when some existing IDs that are not currently in use are available. Also, when two nodes A and B newly join the WSN, as shown in Fig. 2, the common keys of some nodes will be changed. For example, originally the common key of nodes 1 and 5 was $K_{1,1}$. After sensors A and B join the network, the common keys of nodes 1 and 5 becomes $K_{4,5}$. Now, the system cannot work normally.

3 THE PROPSED SCHEME

The SKMaS consists of four working phases: the key pre-distribution, dynamic shared key establishing, key refreshment, and data transmission phases. In the key pre-distribution phase, the KDS generates a $n \times n$ key matrix K, in which the keys are pseudo-random numbers. After that, the KDS assigns these keys to sensors during the deployment of sensor nodes. Before communicating with each other,

(a) KDS assigns each sensor node two sets of keys (the shadowed parts)

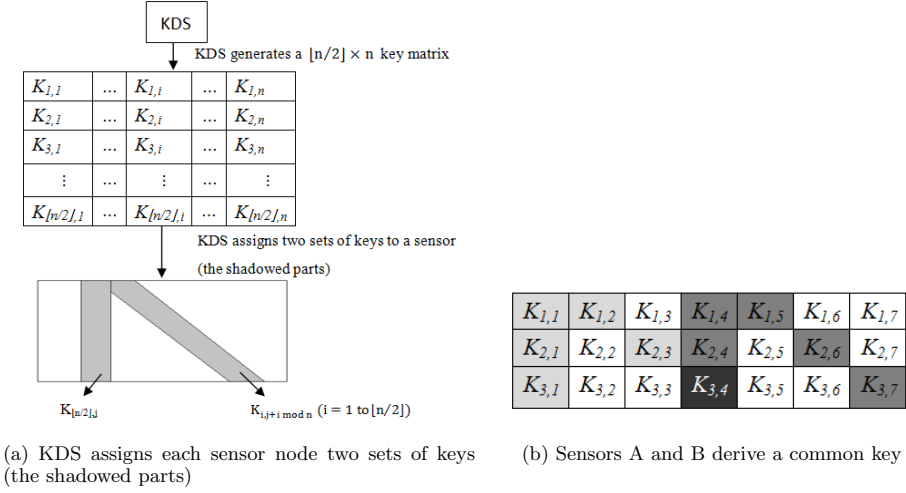(b) Sensors A and B derive a common key

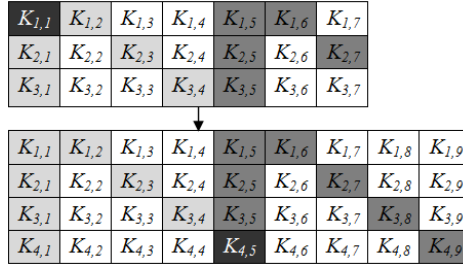Fig. 1: The KDS generates a key matrix and assigns common keys to a sensor

Fig. 2: New sensor nodes A (node 8) and B (node 9) join the network

each pair of sensors needs to identify the CKs (recall communication keys) shared with each other, and then generates the DSK (recall Dynamic Shared Key) in the shared key establishing phase. When the sensor, e.g., m, newly joins the network, the KDS broadcasts the ID (i.e., m), and the CKs generated for m. Now 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 authenticate received messages to see whether they are sent by valid sensors or not.

3.1 Key pre-distribution phase

As stated above, each sensor i is given two sets of keys. The first set, i.e., CKs consisting of all keys collected in row i and column i in the key matrix K (row i and column i together are called key-cross i), is used to perform one-to-one com-

munication between i and another sensor, e.g., j, by using the computed DSK, $1 \leq i,j \leq n, i \neq j$. The other one, i.e., individual key $K_{i,i}$, is the key employed by sensor i to communicate with the KDS where $K_{i,i}$ is the ith element along the diagonal of the key matrix K. The steps of the key pre-distribution phase are as follows.

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

Step 2: the KDS assigns an ID, e.g., i, which is the index of $K_{i,i}$, and the CKs $K_{i,j}$ and $K_{j,i}$ in K to a sensor, $1 \leq i, j \leq n, i \neq j$ (including KDS itself since its ID $= 1$).

Step 3: the KDS generates a system control key $K_{0,0}$ and the individual key $K_{1,1}$, and then sends these keys to all sensor nodes in the system.

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

Fig. 3: 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 $[K_{i,1}, K_{i,2}, \cdots, K_{i,i}, \cdots, K_{i,n}]$ together called key-cross i, are assigned to sensor i.

3.2 Dynamic shared key establishing phase

After the deployment of sensors, when sensor i would like to communicate with sensor j, it sends its own ID, i.e., i, to j. With the two IDs, i (j) can identify the CK, i.e., $K_{i,j}$ and $K_{i,j}$ contained in key-cross i (key-cross j), $1 \leq i,j \leq n, i \neq j$. Before authenticating node j (receiver), node i (sender) generates an authentication code ($Auth$) which contains the result of performing a two dimensional operation $\oplus$ and $+_2$, where $+_2$ is a binary adder which ignores the carry of the most significant bit [30, 31], with $K_{i,j}$, $K_{j,i}$ and a pseudo-random number $rand$ as its parameters where

$$Auth = (K_{i,j} \oplus rand) +_2 K_{j,i} \tag{1}$$

After that, i delivers $rand$ and $Auth$ to j and generates the DSK where

$$DSK = (K_{i,j} \oplus Auth) +_2 (K_{j,i} \oplus rand) \tag{2}$$

On receiving $rand$ and $Auth$, node j retrieves $K_{i,j}$ and $K_{i,i}$ from its own key-cross j, invokes Eq. (1) to calculate $Auth$, denoted by $Auth_c$, and checks to see whether the received $Auth$, denoted by $Auth_r$, is equal to $Auth_c$ or not. If yes, meaning that i is a valid one, j invokes Eq. (2) to generate the DSK.

3.3 Key refreshment phase

When sensor i leaves a WSN, the KDS broadcasts a message (named a $leaving-node$ message) to the remaining sensors. The format of this message is shown in Fig. 4 in which OP_code=0 indicates that this is a leaving-node message.

$$\text{OP\_code} = 0 | \text{node ID} = i | rand | H(K_{0,0} +_2 rand \| K_{1,1} \oplus rand, rand)$$

Fig. 4: The format of a leaving-node message issued by the KDS to announce the leaving of node i

On receiving the message, sensor j, $1 \leq j \leq n, j \neq i$, retrieves keys $K_{0,0}$ and $K_{1,1}$ from its internal file, calculates the authentication code $H(K_{0,0} +_2 rand \| K_{1,1} \oplus rand, rand)$, and checks to see whether $H(K_{0,0} +_2 rand \| K_{1,1} \oplus rand, rand)_c$ is equal to $H(K_{0,0} +_2 rand \| K_{1,1} \oplus rand, rand)_r$ or not, where subscript c denotes calculation and subscript r denotes received. If not, j discards this message. Otherwise, j no longer communicates with i. Now i is available and can be reused.

When a sensor, e.g., d, newly joins the network, it may face two situations: with or without an available previously-used ID in the underlying network.

1. If the situation "with an available previously-used ID" occurs, the KDS assigns an available ID, e.g., i, to d (i.e., $d = i$) and retrieves the corresponding CKs, i.e., key-cross d, from K, $2 \leq i \leq n$.
2. If the situation "without an available previously-used ID" occurs, the KDS generates a new ID d (e.g., $d = n + 1$) and key-cross d, in which $K_{d,i}$ and $K_{i,d}$ are used by d to securely communicate with sensor i for all is, $1 \leq i \leq n$.

Before deploying the new sensor, the system manager has to retrieve key-cross d and node ID d from the KDS and store them into the node. After that, the KDS broadcasts a message, named the $newly-joining$ message, the format of which is shown in Fig. 5, in which the format of k-msg is illustrated in Fig. 6. In Fig. 5, OP_code $= 1$ indicates that this is a newly-joining-node message. Upon receiving this message, sensor i retrieves keys $K_{0,0}$ and $K_{1,1}$ from its internal file, calculates authentication code $H(K_{0,0} \oplus rand \| K_{1,1} +_2 rand, rand)$, and checks to see whether

$H(K_{0,0} \oplus rand \| K_{1,1} +_2 rand, rand)_c$ is equal to $H(K_{0,0} \oplus rand \| K_{1,1} +_2 rand, rand)_r$ or not, where script c denotes calculation and script r denotes received. If not, i discards this message. Otherwise i sequentially searches the ID fields contained in the k-msg field of this message. In the k-msg, a sensor ID, e.g., i, is followed by, $K_{i,i} \oplus K_{d,i}$ and $(K_{0,0} +_2 K_{i,i}) \oplus K_{i,d}$, in which $K_{d,i} and K_{i,d}$ are the CKs needed to be updated by sensor i or added to key-cross i. When ID $= i$ as the head field for sensor i is identified, i decrypts the communication keys conveyed in the two fields following the head field, and accordingly updates its key-cross i.

$$\text{OP\_code} = 1|rand|k\text{-msg}|H(K_{0,0} \oplus rand \| K_{1,1} +_2 rand, rand)$$

Fig. 5: The format of a newly-joining node message broadcast by the KDS to all sensors. In this message, the format of k-msg is shown in Fig. 6.

ID=2	$K_{2,2} \oplus K_{n+1,2}$	$(K_{0,0} +_2 K_{2,2}) \oplus K_{2,n+2}$	$K_{n+1,2} \oplus K_{2,n+1}$
ID=3	$K_{3,3} \oplus K_{n+1,3}$	$(K_{0,0} +_2 K_{3,3}) \oplus K_{3,n+1}$	$K_{n+1,3} \oplus K_{3,n+1}$
⋮			
ID=n	$K_{n,n} \oplus K_{n+1,n}$	$(K_{0,0} +_2 K_{n,n}) \oplus K_{n,n+1}$	$K_{n+1,n} \oplus K_{n,n+1}$

Fig. 6: The format of k-msg, included in a newly-joining-node message (see Fig. 5), contains sensor ID, e.g., i, and the communication keys, $K_{d,i}$, and $K_{i,d}$ needed to be added to the key-cross i by sensor i, $2 \le i \le n$, where $d = n + 1$

For example, in Fig. 6, if ID $= 2$ and $d = n + 1$, the following two fields are $K_{2,2} \oplus K_{n+1,2}$ and $(K_{0,0} +_2 K_{2,2}) \oplus K_{2,n+1}$. Only the valid KDS has the right individual key $K_{2,2}$ and control key $K_{0,0}$ to encrypt the two fields, and only the valid sensor, i.e., sensor 2, which has the two keys, is able to decrypt the two fields. Moreover, after obtaining $K_{n+1,2}$ and $K_{2,n+1}$, sensor 2 compares the $(K_{n+1,2} \oplus K_{2,n+1})$ 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. In our scheme, the addition of d, no matter whether $d = n + 1$ or $2 \le d \le n$, does not change those CKs currently used by all existing sensors.

3.4 Data transmission phase

After completing the authentication between two sensors, the two sensors can communicate with each other by sending a data message, the format of which is shown in Fig. 7, in which OP_code $= 2$ indicates that this is a data message and $d - msg$ is

the data that the sender, e.g., node i, would like to send to the receiver, i.e., node j. When receiving this message, sensor j retrieves the source ID, Destination ID and $(d-msg \oplus Auth) +_2 DSK$ fields, computes the hash value HMAC=H'$(i\|j\|rand\|(d-msg \oplus Auth) +_2 DSK, DSK)$ 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 H'(x, y) is a hash function hashing x with key y. Since only a valid sensor has the right $Auth$ and DSK to produce the correct hash value, if the two hash values are equal, the message is authenticated, meaning that the sensor sending this message is a valid one.

$$\boxed{\begin{array}{c} \text{OP\_code } = 2 \mid \text{SourceID} = i \mid \text{DestinationID} = j \mid rand \mid \\ (d\text{-}msg \oplus Auth) +_2 DSK \mid \text{H}'(i \| j \| rand \| d\text{-}msg \oplus Auth) +_2 DSK, DSK) \end{array}}$$

Fig. 7: The format of a data message, in which $d-msg$ is the data that sensor i would like to send to sensor j

Let x be $(d-msg \oplus Auth) +_2 DSK$. The $d-msg$ can be obtained by decrypting x where

$$d - msg = \begin{cases} (x - DSK) \oplus Auth, & if\, x \geq DSK \\ (x + \overline{DSK} + 1) \oplus Auth, & if\, x < DSK \end{cases} \tag{3}$$

4 SECURITY ANALYSIS

The SKMaS has four features, including:

1. Verifying whether a transmitted message is a legitimate one or not by checking a hash authentication code with a dynamic key [22], e.g., $rand$ (see Figs. (4), (5) and (7)).

2. The OP_code as the head of a transmitted message explicitly indicates the function of this message to improve the efficiency of the following authentication and message processing.

3. The DSK carried in a data message effectively improves the security level of the message since for different communication sessions, DSKs varies due to invoking different $rand$s.

4. A two dimensional operation (i.e., $+_2$ and $\oplus$) invoked by the SKMaS to encrypt/decrypt data messages enhances the security level of the WSN.

In this section, we analyze the security of the transmitted messages and describe how the SKMaS effectively defend against three common attacks, including eavesdropping [23], forgery KDS, and forgery sensor node [24], and show that the SKMaS can effectively prevent a WSN from being attacked by them.

4.1 Security of a message

The newly-joining-node and leaving-node messages issued by the KDS possess a high security mechanism and are discussed as follows.

First, the leaving-node message shown in Fig. 4 is secure. Since the hacker does not have the control key $K_{0,0}$ and KDS's individual key $K_{1,1}$, he/she cannot generate correct hash authentication code $H(K_{0,0} +_2 rand \| K_{1,1} \oplus rand, rand)$.

Second, the newly-joining-node message illustrated in Fig. 5 is more secure than the leaving-node message since this message contains the hash authentication code $H(K_{0,0} \oplus rand \| K_{1,1} +_2 rand, rand)$ and the self-checking code, $K_{n+1,j} \oplus K_{j,n+1}, 2 \leq j \leq n$.

Basically, only the KDS and the intended receiving sensor node j have the individual key $K_{j,j}, 2 \leq j \leq n$, with which the KDS encrypts the newly-joining node d's CKs (i.e., $K_{j,j} \oplus K_{n+1,j}$ and $(K_{0,0} +_2 K_{j,j}) \oplus rand_{j,n+1}$ fields in Fig. 6 where $d=n+1$) and j decrypts the CKs so as to obtain the correct self-checking code $K_{n+1,j} \oplus K_{j,n+1}$. Even if one of the sensor nodes, e.g., m, was captured by the hacker, with the parameters that m has, the hacker cannot correctly generate other nodes' self-checking codes. We then dare to say that a newly-joining-node message is well protected.

Lemma 1. Let $K_{0,0}, K_{j,j}, K_{j,n+1}$ and $K_{n+1,j}$ be all n bits long, $2 \leq j \leq n$. The probability p of correctly generating the self-checking code $K_{n+1,j} \oplus K_{j,n+1}$ shown in Fig. 6 by the hacker is $p = \frac{1}{2^n}$.

Proof.
To correctly generates the self-checking code $K_{n+1,j} \oplus K_{j,n+1}$, the hacker needs to correctly decrypt the keys $K_{j,j} \oplus K_{n+1,j}$ and $(K_{0,0} +_2 K_{j,j}) \oplus K_{j,n+1}$ to obtain the pseudo-random keys $K_{j,n+1}$ and $K_{n+1,j}$. However, both $K_{0,0}$ and $K_{j,j}$ are unknown to the hacker. The probability of correctly generating $K_{j,j} \oplus K_{n+1,j}$ and $(K_{0,0} +_2 K_{j,j}) \oplus K_{j,n+1}$ by the hacker is $\frac{1}{2^n}$. In other words, when a sensor j receives an illegal newly-joining-node message broadcasted by the hacker, the probability p with which sensor node j correctly decrypt the corresponding portion of the message, i.e., ID $= i \| K_{j,j} \oplus K_{n+1,j} \| (K_{0,0} +_2 K_{j,j}) \oplus K_{j,n+1} \| K_{n+1,j} \oplus K_{j,n+1}$, to obtain the correct values of $K_{n+1,j}$ and $K_{j,n+1}$ is $\frac{1}{2^n}$.

Hence, the probability with which the hacker correctly generates the self-checking code, is $\left(\frac{1}{2^n}\right) \times \left(\frac{1}{2^n}\right) \times 2^n = \frac{1}{2^n}$, where 2^n is the number of the possible value-combinations of each of $K_{n+1,j}$ and $K_{j,n+1}$
$\square$

Lemma 2. Let $K_{0,0}, K_{j,j}, K_{j,n+1}$ and $K_{n+1,j}$ be n bits long, $2 \leq j \leq n$. The prob-

ability p of recovering the correct value of $K_{j,j}$ from the corresponding portion ID= $j \mid K_{j,j} \oplus K_{n+1,j} \mid (K_{0,0} +_2 K_{j,j}) \oplus K_{j,n+1} \mid K_{n+1,j} \oplus K_{j,n+1}$ conveyed in the illegally intercepted newly-joining-node message on one trial is $p = \frac{1}{2^n}$.

Proof.

On receiving the corresponding portion of the newly-joining-node message, the hacker may guess a pair of keys (X, Y) such that $X \oplus Y = (K_{n+1,j} \oplus K_{j,n+1})_r$ where subscript r denotes that the value of $X \oplus Y$ is retrieved from the message. Continuously, he/she may try to obtain the individual key $K_{j,j}$ by performing $(K_{j,j})_c = (K_{n+1,j} \oplus K_{j,n+1})_r \oplus X$ and $(K_{0,0} +_2 K(j,j))_c = (K_{n+1,j} \oplus K_{j,n+1})_r \oplus Y$, where subscript c denotes the value calculated by the hacker. However, without knowing $K_{j,j}$ and $K_{0,0}$, the hacker cannot verify whether the calculated values of $(K_{j,j})_c$ and $(K_{0,0}+_2K_{j,j})_c$ are correct or not. That is, even though the hacker receives the newly-joining-node message and retrieves the portion ID $= j \mid K_{j,j} \oplus K_{n+1,j} \mid (K_{0,0} +_2 K_{j,j}) \oplus K_{j,n+1} \mid K_{n+1,j} \oplus K_{j,n+1}$, there is no way for the hacker to make sure that the obtained $K_{j,j}$ is correct or not, except by a blind guess. Hence, the probability p of recovering the correct value of $K_{j,j}$ from the portion of an illegally intercepted newly-joining-node message on one trial is $p = \frac{1}{2^n}$.

Furthermore, the data message m delivered between sensor node i and sensor node j, as shown in Fig. 7, is protected by the pseudo-random variables $rand$, $Auth$ and DSK, which are themselves different in different times of communication.

Arguments of above two paragraphs contribute two security mechanisms for a data message.   □

1. The hash authentication code H'$(i\|j\|rand\|(d - msg \oplus Auth) +_2 DSK, DSK)$ contained in a data message possesses three security functions, including authentication [25], integrity [26], and non-repudiation [27].
2. $d - msg$ shown in Fig. 7 is well protected by $Auth$, DSK, and two-dimensional operation $\oplus$ and $+_2$. Only sensor node i and sensor node j have the communication keys $K_{i,j}$ and $K_{j,i}$, by which they can correctly encrypt/decrypt the $d - msg$.

From the above analyses, we can see that the newly-joining-node message shown in Fig. 5 and the data message illustrated in Fig. 7 are well protected by the SKMaS.

4.2 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, illegal users cannot decrypt messages protected by DSK derived from $rand$, $K_{i,j}$ and $K_{j,i}$ (see Eqs.(2) and (3)). In this study, different messages are

dynamically encrypted by different pseudo-random keys, i.e., *rand*s, thus having a higher security level than that protected by static keys [28]. In other words, messages delivered in the data transmission phase are secure. So the eavesdropping attack does not work.

4.3 Forgery KDS attack

A forgery KDS may send faked 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 $K_{i,i}$, $2 \leq i \leq n$, which is only known to sensor i and the KDS, and is used to encrypt messages and authenticate the integrity of messages delivered between i and KDS. Therefore, only the valid KDS has the right $K_{i,i}$ and only i can correctly use it to decrypt the messages issued by the KDS, meaning that the SKMaS can effectively defend the forgery KDS attack.

4.4 Forgery sensor node attack

If a hacker, e.g., b', disguising itself as the valid sensor b, sends a data message to c, since b' does not have $K_{b,c}$ and $K_{c,b}$, the data message cannot pass the authentication performed by c (see Fig. 7). Thus b' is incapable of identifying the right DSK for further interacting with sensor c. Also, a faked node cannot decrypt messages issued by a valid one because it does not own the right DSK.

5 SIMULATION AND PERFORMANCE

Our experimental environment is developed on two identical desktop computers equipped with an Intel i5-3450 at 3.1Ghz 16Gb of memory and running the Windows 7 Operating System with Java JDK 7u21. The wireless environment is IEEE.802.11g with 54Mbps as its transmission speed. The expression generation times of the dynamic shared key establishing, key refreshment and data transmission phases on different key lengths are shown in Table. 1. In this table, the time consumed to generate the pseudo-random parameter *rand* is long, making it around ten times the time required to generate those expressions excluding the *rand*. The average generation times of the expressions excluding the generation of the *rand* on 256, 512 and 1024 bits are around 1.5, 3 and 6 ms, respectively. If the generation of the *rand* is required, the times on 256, 512, and 1024 bits are around 15, 30, and 55 ms, respectively, showing that the proposed scheme has low computation overhead. Since the experiment was performed on desktop computers with Wi-Fi, the times required by employing real sensors equipped with Zigbee as their wireless communication protocol need to be multiplied by 100 when they are tested in real sensors.

Table 1: The expression generation times in the dynamic shared key establishing, key refreshment and data transmission phases

Dynamic shared key establishing phase				
Expression length	256 bits (ms)	512 bits (ms)	1024 bits (ms)	Note
$Auth = (K_{i,j} \oplus rand) +_2 K_{j,i}$ (see Eq. (1))	14.931	31.176	53.845	Retreving $K_{i,j}$ and $K_{j,i}$ from key-cross i or K, generating a $rand$, and performing $+_2$ and $\oplus$
$DSK = (K_{i,j} \oplus Auth) +_2 (K_{j,i} \oplus rand)$ (see Eq. (2))	1.757	3.284	6.953	Reusing $K_{i,j}$, $K_{j,i}$, $Auth$ and $rand$ retrieved and produced in the generation process of $Auth$, and performing $+_2$ and $\oplus$
Key refreshment phase				
H($K_{0,0}$ $+_2$ $rand\|K_{1,1}$ $\oplus$ $rand\|K_{1,1}$ $+_2$ $rand, rand$) (see Fig. 4)	15.331	31.712	58.161	Retreving $K_{0,0}$ and $K_{1,1}$, generating a new $rand$, and performing $+_2$, $\oplus$, $\|$ and H(,)
H($K_{0,0}\|K_{1,1}$ $+_2$ $rand$ $+_2$ $rand, rand$) (see Fig. 5)	15.364	29.031	57.228	Retreving $K_{0,0}$ and $K_{1,1}$, generating a new $rand$, and performing $+_2$, $\oplus$, $\|$ and H(,)
Data transmission phase				
$(d - msg \oplus Auth) +_2 DSK$ (see Fig. 7)	1.615	2.916	5.833	Reusing $Auth$, and DSK, generating $d - msg$, and performing $+_2$ and $\oplus$
H'($i\|j\|rand\|(d - msg \oplus Auth) +_2 DSK, DSK$) (see Fig.7)	14.371	28.17	53.632	Reusing $(d - msg \oplus Auth) +_2 DSK$, generating a new $rand$, and performing $+_2$, $\oplus$, $\|$ and H(,)

The message transmission times are shown in Table 2, in which the longest time was performing key refreshment. In this phase, k-msg is a variable, the size of which varies depending on the number of invoked senors, i.e., n, in the underlying WSN. For example, if $n = 1000$, the time required to send the key refreshment message (Figs. 5 and 6) on key length $=1024$ bits is $0.021 + 0.057 \times 1000 = 57.021ms$. This is still acceptable.

Table 2: The message transmission times of the key refreshment and data transmission phases.

Key length	256 bits (ms)	512 bits (ms)	1024 bits (ms)
The key refreshment: The leaving-node message (see Fig.4)	0.007	0.012	0.021
The key refreshment: The newly-joining-node message (see Figs. 5 and 6)	0.007+0.014(n-1)	0.012+0.029(n-1)	0.021+0.057(n-1)
The data message: A data message (Fig.7)	0.012	0.022	0.041

6 CONCLUSIONS AND FUTURE STUDIES

In this paper, we design and analyze a square key matrix 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 since a sensor network does not have available previously-used IDs, adding extra sensor nodes will change the DSKs used by other nodes and may aggravate or even crash the whole system. We also evaluate and show that the proposed system can effectively defend from three common attacks. The system enhances the security and resiliency of the sensor networks without conducting tremendous amount of computation and complicated cryptographic techniques.

In the future, we would like to improve the reliability and derive working model 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 pseudo-random number keys illustrated in Fig. 6. In other words, we only need to invoke a function instead of issuing a big message containing k-msg (see Fig. 5) or n authentication messages for message authentication. These constitute our future studies.

7 ACKNOWLEDGEMENTS

8 REFERENCE

1. Leu F.Y.: Emerging Security Technologies and Applications. In: Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, September 2011, vol. 2, no. 3, pp. 1-3.

2. Tanveer A. Zia and Albert Y. Zomaya: A Lightweight Security Framework for Wireless Sensor Networks. In: Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, vol. 2, no. 3, September 2011, pp. 53-73.

3. Durisic, M.P. and Tafa, Z., Dimic, G. and Milutinovic, V.: A Survey of Military Applications of Wireless Sensor Networks. In: Mediterranean Conference on Embedded Computing, June 2012, pp. 196-199.

4. Bekmezci, I. and Alagoz, F.: Energy Efficient, Delay Sensitive, Fault Tolerant Wireless Sensor Network for Military Monitoring. In: IEEE International Conference on Sensors Applications Symposium, February 2008, pp. 172-177.

5. Ranasingha, M.C. and Murthi, M.N. and Premaratne, K. and Fan, X.: Transmission Rate Allocation in Multi-Sensor Target Tracking. In: IEEE International Conference on Acoustics, Speech and Signal Processing, March/April 2008, pp. 3021-3024.

6. Chen Y.M, Shen W., Huo H.W. and Xu Y.Z.: A Smart Gateway for Health Care System Using Wireless Sensor Network. In: International Conference on Sensor Technologies and Applications, July 2010, pp. 545-550.

7. Johansson, A., Shen, W. and Xu, Y.: An ANT Based Wireless Body Sensor Biofeedback Network for Medical E-Health Care. In: International Conference on Wireless Communications, Networking and Mobile Computing, 2011, pp. 1-5.

8. Kong Y.F. and Jiang P.: Development of Data Video Base Station in Water Environment Monitoring Oriented Wireless Sensor Networks. In: International Conference on Embedded Software and Systems Symposia, July 2008, pp. 281-286.

9. Freeman, J.D. and Simi, S.: Remote Monitoring of Indoor Environment using Mobile Robot Based Wireless Sensor Network. In: International Conference on Computer Science and Education, August 2011, pp. 1080 – 1084.

10. Elmagid, A.A.A., Ramadan, R.A., El-Ghanam, S.M., Al-tabbakh, S.M., Kamh, S.A. and Marie, M.: Radiation Detection Based Heterogeneous Wireless Sensor Network. In: International Conference on Engineering and Technology, October 2012, pp. 1-6.

11. Jian B., Luo C.Y., Guo Y.H. and Li W.: A New Key Pre-distribution Scheme for Wireless Sensor Networks. In: International Symposium on Information Engineering and Electronic Commerce, May 2009, pp. 333-336.

12. Sharmila Deva Selvi1 S., Sree Vivek1 S., Vivek Krishna Pradhan., and Pandu Rangan C.: Efficient Certificateless Online/Offline Signature. In: Journal of Internet Services and Information Security, November 2012, vol 2, issue 3/4, pp. 77-92.

13. Claycomb W., Shin D.: Extending Formal Analysis of Mobile Device Authentication. In: Journal of Internet Services and Information Security, May 2011, vol 1, issue 1, pp. 86-102.

14. Wuu L.C., Hung C.H. and Chang C.M.: Quorum-based Key Management Scheme in Wireless Sensor Networks. In: International Conference on Ubiquitous Information Management and Communication, 2012, No. 15.

15. Huang Y.L. and Leu F.Y.: Constructing a Secure Point-to-Point Wireless Environment by Integrating Diffie-Hellman PKDS RSA and Stream Ciphering for Users Known to Each Other. In: Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, vol. 2, no. 3, September 2011, pp. 96-107.

16. Castiglione, A., Cattaneo, G., Cembalo, M., De Santisa, A., Faruolo, P., Petagna, F., Petrillo, U.F.: Engineering a secure mobile messaging framework. In: Computers and Security, vol. 31, September 2012, pp. 771-781.

17. Castiglione, A., Cattaneo, G., Maio, G.D., Petagna, F.: Secure End-to-End Communication over 3G Telecommunication Networks. In: International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2011, pp. 520-526.

18. De Santis, A., Castiglione, A., Cattaneo, G., Cembalo, M., Petagna, F., Petrillo, U.F.: An Extensible Framework for Efficient Secure SMS. In: International Conference on Complex, Intelligent and Software Intensive Systems, February 2010, pp. 15-18.

19. Hwu J.S., Hsu S.H, Lin Y.B, Chen R.J.: End-to-end security mechanisms for SMS. In: International Journal of Security and Networks, 2006, vol 1, no. 3/4, pp. 177-183.

20. Agoyi, M., Seral, D.: An Asymmetric Encryption Approach. In: International Conference on Wireless and Mobile Communications, September 2010, pp. 448-452.

21. Cheng, Y. and Agrawal, D.P.: Efficient Pairwise Key Establishment and Management in Static Wireless Sensor Networks. In: IEEE International Conference on Mobile Ad hoc and Sensor Systems, 2005, pp. 544-550.

22. He X.B., Niedermeier, M., Meer, H.D.: Dynamic key management in wireless sensor networks: A survey. In:Journal of Network and Computer Applications, 2013, vol. 36, issue 2, pp. 611-622.

23. Huang, X., Ahmed, M. and Sharma, D.: Timing Control for Protecting from Internal Attacks in Wireless Sensor Networks. In: International Conference on Information Networking, February 2012, pp. 7-12.

24. Li Z. and Ravishankar, C.V.: A Fault Localized Scheme for False Report Filtering in Sensor Networks. In: International Conference on Pervasive Services, July 2005, pp. 59-69.

25. Chen R.: Research on Security Authentication of Hierarchy Distributed Wireless Sensor Network. In: International Conference on Computer and Automation Engineering, February 2010, pp. 275-278.

26. Stephens, B., Tectura, Corp. and Bellevue, W.A.: Security Architecture for Aeronautical Networks. In: Digital Avionics Systems Conference, October 2004, vol. 2.
27. Xue H., Zhang H., Qing S. and Yu R.: Automated Design of Non-Repudiation Security Protocols. In: International Conference on Wireless Communications, Networking and Mobile Computing, September 2007, pp. 2318-2321.
28. Liu, D. and Ning, P.: Improving Key Pre-Distribution with Deployment Knowledge in Static Sensor Networks. In: ACM Transactions on Sensor Networks, 2005, vol.1, pp. 204-239.
29. Castiglione, A., Marco Cepparulo, M., De Santis, A., Palmieri, F.: Towards a Lawfully Secure and Privacy Preserving Video Surveillance System. In: International Conference on E-Commerce and Web Technologies, 2010, pp. 73-84.
30. Huang Y.L., Leu F.Y., Wei K.C.: A Secure Communication over Wireless Environments by using a Data Connection Core. In: International Journal of Mathematical and Computer Modeling, 2013, vol. 58, issue 5/6, pp. 1459-1474.
31. Huang Y.L., Dai C.R., You I. and Leu F.Y.: A Secure Data Encryption Method Employing a Sequential-Logic Style Mechanism, Three-Dimensional Operation and Dynamic Transition Box. In:International Journal of Web and Grid Services.