Quorum-based Key Management Scheme in Wireless Sensor Networks

Lih-Chyau Wuu[1], Chi-Hsiang Hung[2] and Chia-Ming Chang[3]
[1,2,3] Graduate School of Computer Science and Information Engineering,
National Yunlin Unviersity of Science and Technology, Yunlin, 640 - Taiwan
[e-mail: {wuulc, g9510802, g9717710}@yuntech.edu.tw]
*Corresponding author: Lih-Chyau Wuu

Received May 29, 2012; revised July 31, 2012; accepted August 17, 2012
published September 26, 2012

Abstract

To ensure the security of wireless sensor networks, it is important to have a robust key management scheme. In this paper, we propose a Quorum-based key management scheme. A specific sensor, called as key distribution server (KDS), generates a key matrix and establishes a quorum system from the key matrix. The quorum system is a set system of subsets that the intersection of any two subsets is non-empty. In our scheme, each sensor is assigned a subset of the quorum system as its pre-distributed keys. Whenever any two sensors need a shared key, they exchange their IDs, and then each sensor by itself finds a common key from its assigned subset. A shared key is then generated by the two sensors individually based on the common key. By our scheme, no key is needed to be refreshed as a sensor leaves the network. Upon a sensor joining the network, the KDS broadcasts a message containing the joining sensor ID. After receiving the broadcast message, each sensor updates the key which is in common with the new joining one. Only XOR and hash operations are required to be executed during key update process, and each sensor needs to update one key only. Furthermore, if multiple sensors would like to have a secure group communication, the KDS broadcasts a message containing the partial information of a group key, and then each sensor in the group by itself is able to restore the group key by using the secret sharing technique without cooperating with other sensors in the group.

Keywords: Wireless sensor network, key management, quorum, group key

A preliminary version of this paper appeared in ACM ICUIMC 2012, February 14-18, Kuala Lumpur, Malysia. The research was supported by the National Science Council, Taiwan, under contract number NSC 100-2221-E-224-054.
The authors would like to thank the anonymous reviewers for their valuable comments. The authors would also like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research.
http://dx.doi.org/10.3837/tiis.2012.09.027

1. Introduction

Recent advances in micro-electro-mechanical devices and radio, signal processing technologies have made wireless sensor networks (WSNs) being well developed in a wide variety of applications, such as battlefield surveillance, industrial process monitoring and control, healthcare monitoring, home automation, traffic control, etc. However, some applications may require certain security mechanisms to protect the integrity of passing data from modifications, guard for confidentiality of communication from electronic eavesdrop, and verify message originator. To provide the above security-enhanced services for wireless sensor networks, it is necessary to have a robust key generation and distribution scheme.

Considering the limited power, data processing capacity and memory storage of sensors, traditional key generation and distribution schemes are inapplicable for wireless sensor networks [1][2][3][4][5][6][7]. To solve the key distribution problem for wireless sensor networks, several key pre-distribution schemes have been proposed in the literatures [2][3][4][5][6][7][8][9][10][11][12]. The basic idea is that keys would have to be installed in sensors before deployment. A simple key pre-distribution approach is to assign each sensor n-1 keys, where n is the number of sensors in the network, and each key is known to only one other sensor. However, this approach is impractical as n is large or the network is highly dynamic.

Eschenauer et al. [8] proposed a random key pre-distribution scheme in 2002. In this scheme, a very large size symmetric key pool (e.g., 2^{17} - 2^{20} keys) is generated offline at first, and then each sensor randomly selects a set of keys from the generated key pool as its pre-distributed keys. After that, the sensors are randomly deployed into an interested terrain. After the deployment, each sensor broadcasts its stored key information to its one-hop neighbors. Since all the keys are randomly selected from the same key pool, it is quite possible that two neighboring sensors have some overlapped keys. If two sensors have a common key, they can use it as their shared key directly. Otherwise, a path-key establishment procedure is triggered, which could generate a path-key between the two communicating sensors under some other intermediate sensor's participation. However, that intermediate sensors involve the path-key establishment procedure between the two communicating sensors not only degrades the network security, but also produces additional communication and computational overhead. Furthermore, the scheme can not achieve fully key connectivity since it is possible that two sensors can not derive a shared key even the path-key establishment procedure [8] had been trigged.

To achieve the fully key connectivity, the key pre-distribution scheme of Cheng et al. [4] uses one $\sqrt{n}\times\sqrt{n}$ matrix only, where n is the network size. Each sensor is loaded with a row and a column of the matrix randomly. After deployment, two adjacent sensors exchange their row and column numbers to find out the keys being shared in common by them. The shared key is then derived from the common keys. In this scheme, if there were two sensors with the same row or column of the key matrix, and one of them was captured by an attacker, the attacker can compromise the network security by using the common keys. Such an attack is called as the node capture attack [4].

Based on Blom's method [13], Chien et al. [6] proposed a key pre-distribution scheme by using two $n\times n$ matrices: a public matrix M and a secret symmetric random matrix D. The matrices M and D are used to compute a symmetric matrix $K=(DM)^TM$, where $K_{i,j}=K_{j,i}$. At the key pre-distribution phase, each sensor i is loaded with col_i (the ith column of matrix M),

which is used as public information; and each sensor is also loaded with row_i (the ith row of matrix $(DM)^T$), which is kept secret. After deployment, each sensor i broadcasts its column instances of M (i.e., col_i) to establish a shared key with all its neighbors j by computing pair-wise key $K_{i,j}=row_i \times col_j = row_j \times col_i$. After that, each sensor erases all the pre-loaded parameters (col_i and row_i) from its memory to prevent the possible risk caused by the node capture attack. The erase operation makes the scheme to be inappropriate for dynamic sensor networks.

In this paper, we propose a Quorum-based key pre-distribution scheme which not only achieves the fully key connectivity and resists against node capture attack, but also can be applied to dynamic sensor networks. Our scheme uses one $\lfloor n/2 \rfloor \times n$ matrix and generates a quorum system from the matrix, where n is odd. A quorum system is a set of subsets with the property that the intersection of any two subsets is non-empty. Our scheme assigns each sensor a subset of the quorum system as pre-distributed keys. That guarantees a common key can be found out between any two sensors after they exchange their IDs. Furthermore, such a key assignment makes any pre-distributed key held by two sensors at most. By this feature, no key is needed to be refreshed as a sensor leaves the network. Upon a sensor joining the network, KDS broadcasts the joining sensor's ID, and then each sensor by itself is able to update the key which is in common with the new joining one. The low communication overhead and low computational overhead makes our scheme appropriate to dynamic sensor networks. Moreover, for supporting secure group communications, we require that the KDS is responsible to generate a group key if necessary. The group key is regarded as a secret and is divided into two shadows by the secret sharing technique [18]. By our scheme, a sensor in the group by itself is able to derive one shadow of the secret, and the KDS broadcasts the other shadow. After that, the group key can be restored by each sensor in the group without cooperating with other sensors.

The rest of the paper is organized as follows. Section 2 gives a brief introduction of the quorum system. Section 3 describes the proposed key management scheme. Section 4 and section 5 depicts security analysis and performance evaluation. Finally is the conclusion.

2. Quorum System

A quorum system [14][15][16][17] is a set system $S = \{S_1, S_2, \ldots, S_m\}$, where each S_i is a subset of a universe set $U = \{U_1, U_2, \ldots, U_N\}$, and the quorum system must satisfy the following properties:

- Non-empty intersection property: Any two subsets have a non-empty intersection.
  $\forall S_i, \forall S_j; S_i, S_j \in S; S_i \cap S_j \neq \phi$
- Minimality property: No subset is a proper subset of another subset.
  $\forall S_i, \forall S_j; S_i, S_j \in S; S_i \not\subset S_j$

For example, given a universe set $U = \{U_1, U_2, U_3\}$, a set system $S = \{S_1, S_2, S_3\}$ where $S_1 = \{U_1, U_2\}$, $S_2 = \{U_2, U_3\}$, $S_3 = \{U_1, U_3\}$, S is a quorum system since it has the non-empty intersection property and the minimality property.

3. Quorum-based Key Management Scheme

In our scheme, there exists a special sensor, called as key distribution server (KDS). It is assumed that the KDS is trusted and the number of sensors in the network is less than or equal

to n, where n is odd. Our scheme consists of four phases: key pre-distribution, shared key establishment, key refreshment and group key establistment. Table 1 shows the notations used in this paper. Note that the $i+j$ mod n operation in the paper is defined to have the set of residues $\{1, 2, \ldots, n\}$. That is, $i+j$ mod $n=i+j$ when $i+j \leq n$; $i+j$ mod $n=(i+j)-n$ when $i+j > n$.

Table 1. Notations

Notation	Description
$K_{i, j}$	The i-row and j-column of key matrix K
$CK_{A\text{-}B}$	Common key being found in the pre-distributed keys of sensors A and B
$SK_{A\text{-}B}$	Shared key being established by sensor A and sensor B
$H()$	Hash function
GK	Group key
$i+j$ mod n	$i, j \in \{1, 2, \ldots, n\}$, $i+j$ mod $n=i+j$ if $i+j \leq n$; $i+j$ mod $n=(i+j)-n$ if $i+j > n$

3.1 Key Pre-distribution Phase

Fig. 1 shows that, KDS generates a $\lfloor n/2 \rfloor \times n$ matrix K and establishes a quorum system $S = \{S_1, S_2, \ldots, S_n\}$ from the matrix K, where each S_j contains one entire column j of the matrix K and $\lfloor n/2 \rfloor$ elements out of each of the following $\lfloor n/2 \rfloor$ columns after the column j. It means that each S_j contains n-1 elements: $K_{i, j}$ and $K_{i, j+i \bmod n}$ (i=1 to $\lfloor n/2 \rfloor$). After that, KDS distributes S_j to the sensor whose ID is j through a secure channel. For example, given a key matrix $K_{3\times7}$ as **Fig. 2**, Sensor 4 will hold the pre-distributed keys of the 4^{th} column $\{K_{1, 4}, K_{2, 4}, K_{3, 4}\}$ plus three keys $\{K_{1, (4+1)}, K_{2, (4+2)}, K_{3, (4+3)}\}$ of the matrix $K_{3\times7}$. The details of the key pre-distribution phase are as follows:

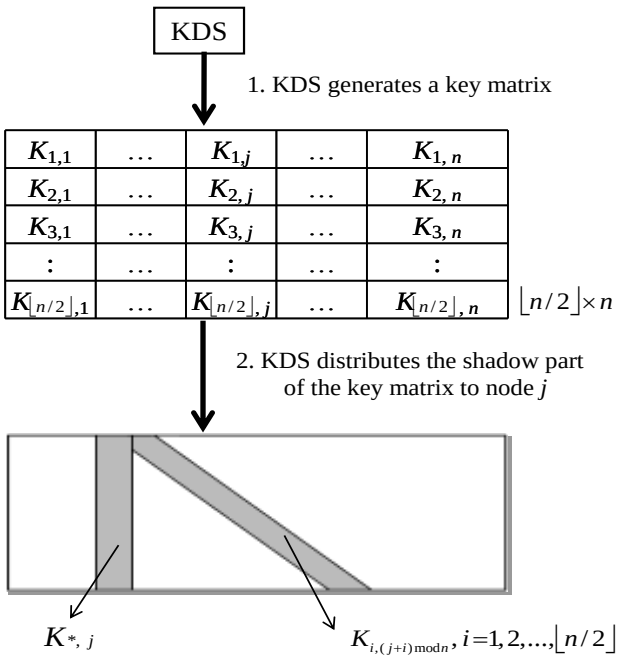

Fig. 1. Key pre-distribution phase

StepP1 : KDS generates a matrix $K_{\lfloor n/2 \rfloor \times n}$ and a random number r, and then computes $H^l(r)$ which will be used for message authenticity at the key refreshment phase, where l is an integer and $H()$ is a secure hash function.

StepP2 : KDS assigns the quorum subset S_j to sensor j, where S_j contains the j^{th} column of K (denoted as $K_{*,j}$) and $K_{i, j+i \bmod n}$ (i=1 to $\lfloor n/2 \rfloor$), and distributes S_j through a secure channel. Each sensor j stores S_j and $H^l(r)$ in its memory.

$K_{1,1}$	$K_{1,2}$	$K_{1,3}$	$K_{1,4}$	$K_{1,5}$	$K_{1,6}$	$K_{1,7}$
$K_{2,1}$	$K_{2,2}$	$K_{2,3}$	$K_{2,4}$	$K_{2,5}$	$K_{2,6}$	$K_{2,7}$
$K_{3,1}$	$K_{3,2}$	$K_{3,3}$	$K_{3,4}$	$K_{3,5}$	$K_{3,6}$	$K_{3,7}$

3×7

Fig. 2. An example of pre-distribution keys of sensor 4

3.2 Shared Key Establishment Phase

After sensors have been deployed, any two sensors A and B can derive a common key based on their IDs, and then establish a shared key to secure their communication channel. Recall that sensor A holds n-1 keys: the entire column A of the matrix K ($K_{*, A}$) and $K_{i, A+i \bmod n}$ (i=1 to $\lfloor n/2 \rfloor$). That indicates the sensor with ID (A+i) also holding the key $K_{i, A+i \bmod n}$ since the entire column (A+i) keys are assigned to the sensor with ID (A+i). The same reason for the key $K_{i, A}$ is held by both of sensor A and the sensor with ID (A-i). This is the reason why our scheme can achieve the fully key connectivity since any two sensors can have a common key from their pre-distribution keys. **Fig. 3** illustrates the exchanged messages at the shared key establishment phase, and the process of establishing a shared key is depicted as follows.

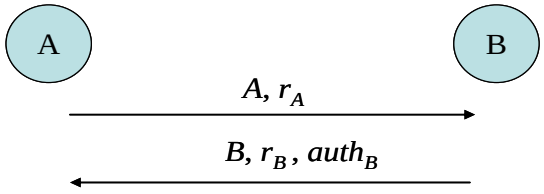

Fig. 3. Shared key establishment phase

StepS1 : Before having a secure channel between sensor A and sensor B, sensor A sends it's ID and a random number r_A to sensor B.

StepS2 : Upon receiving the message $\{A, r_A\}$, sensor B generates a random number r_B and finds out the common key $CK_{A\text{-}B}$ by the IDs A and B. As shown in **Fig. 4**, there are two cases to be considered for finding the $CK_{A\text{-}B}$:

Case1: When $|A\text{-}B| > \lfloor n/2 \rfloor$, sensor A and sensor B have a common key $CK_{A\text{-}B}=K_{n\text{-}|A\text{-}B|, \min(A, B)}$.

Case2: When $|A\text{-}B| \le \lfloor n/2 \rfloor$, sensor A and sensor B have a common key $CK_{A\text{-}B}=K_{|A\text{-}B|, \max(A, B)}$.

The shared key is computed as $SK_{A\text{-}B} = CK_{A\text{-}B} \oplus r_A \oplus r_B$. After that, sensor B sends $\{B, r_B, auth_B\}$ to A, where $auth_B = H(SK_{A\text{-}B} || r_A || r_B)$. The $auth_B$ is used as the message authenticator.

StepS3 : After receiving the message$\{B, r_B, auth_B\}$, sensor A gets the common key $CK_{A\text{-}B}$ based on the IDs as illustrated at Step S2, and then computes the shared key $SK_{A\text{-}B} = CK_{A\text{-}B} \oplus r_A \oplus r_B$, and $auth_A = H(SK_{A\text{-}B} || r_A || r_B)$. After that, sensor A checks whether the $auth_B$ of the message is equal to $auth_A$ to validate the authenticity of sensor B and make sure that they generate the same $SK_{A\text{-}B}$.

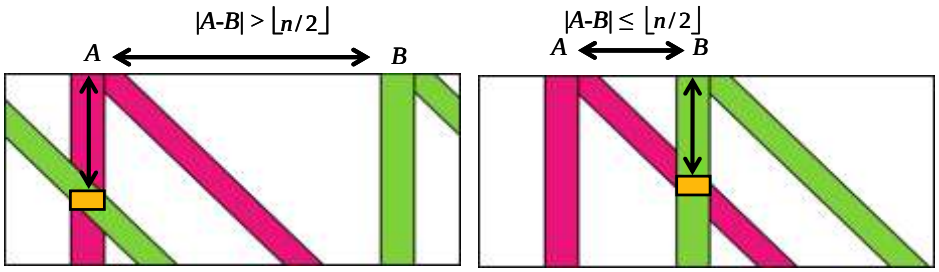

Fig. 4. Finding out a common key for sensor A and B

By the same matrix of **Fig. 2**, **Fig. 5** illustrates that sensor 4 and sensor 6 find out their common key to be $K_{2,6}$ after exchanging their IDs 4 and 6. It is because that |4-6|=2<3, by Case 2, sensor 4 and sensor 6 get the same common key $CK_{4\text{-}6} = K_{|4\text{-}6|, \max(4,6)} = K_{2,6}$.

Node 4

$K_{1,1}$	$K_{1,2}$	$K_{1,3}$	$K_{1,4}$	$K_{1,5}$	$K_{1,6}$	$K_{1,7}$
$K_{2,1}$	$K_{2,2}$	$K_{2,3}$	$K_{2,4}$	$K_{2,5}$	$K_{2,6}$	$K_{2,7}$
$K_{3,1}$	$K_{3,2}$	$K_{3,3}$	$K_{3,4}$	$K_{3,5}$	$K_{3,6}$	$K_{3,7}$

Node 6

$K_{1,1}$	$K_{1,2}$	$K_{1,3}$	$K_{1,4}$	$K_{1,5}$	$K_{1,6}$	$K_{1,7}$
$K_{2,1}$	$K_{2,2}$	$K_{2,3}$	$K_{2,4}$	$K_{2,5}$	$K_{2,6}$	$K_{2,7}$
$K_{3,1}$	$K_{3,2}$	$K_{3,3}$	$K_{3,4}$	$K_{3,5}$	$K_{3,6}$	$K_{3,7}$

Fig. 5. Finding out a common key $K_{2,6}$ for sensor 4 and 6

***Lemma* 1**. Any key in the matrix $K_{\lfloor n/2 \rfloor \times n}$ is held by two sensors at most.

Proof: By our assignment, a key $K_{i,j}$ is held by the sensor j and senor j-i. Assume that there is another sensor x ($x \neq j$ and $x \neq j$-i) also holds the key $K_{i,j}$. Recall that the keys assigned to the sensor x are $K_{*,x}$ and $K_{l, x+l \bmod n}$ (l=1 to $\lfloor n/2 \rfloor$). It is obvious that none of $K_{*,x}$ is $K_{i,j}$ since $x \neq j$. If some key $K_{l, x+l \bmod n} = K_{i,j}$, then $l=i$, $x+l = x+i = j$, that is $x = j$-i. It is contradiction to $x \neq j$-i. Thus, the lemma holds.

3.3 Key Refreshment Phase

Lemma 1 shows that any key in matrix $K_{\lfloor n/2 \rfloor \times n}$ is only held by two sensors at most. In this case, while a sensor leaves the network, KDS only needs to announce the leaving sensor's ID, and no one needs to refresh its pre-distributed keys since the keys held by the leaving sensor will never be used to generate a shared key by the remaining sensors in the network. Our scheme can resists against node capture attack by regarding the captured sensor as a leaving one. The

sensors never establish a shared key with a leaving sensor, even if an attacker spoofed some sensor's ID, it could not derive a correct shared key based on the captured keys.

When a sensor joins the network, KDS assigns an ID and a quorum subset of the updated pre-distributed keys, and distributes the keys through a secure channel, and then broadcasts the joining sensor's ID to require the other sensors to update the key in common with the new joining one. The key refreshment phase is as follows:

StepR1 : KDS assigned an ID A to the new joining sensor.

StepR2 : KDS computes $H^{l-m}(r)$, where m is an integer and $m<l$. KDS assigns the keys $K_{i,A}$, $K_{i,(A+i) \bmod n}$, (i=1 to $\lfloor n/2 \rfloor$) to the new joining sensor A through a secure channel, where $K_{i,A} = H(K_{1,A-i} \oplus K_{2,A-i} \oplus \ldots \oplus K_{\lfloor n/2 \rfloor,A-i} \oplus H^{l-m}(r))$, and $K_{i,(A+i) \bmod n} = H(K_{1,A+i} \oplus K_{2,A+i} \oplus \ldots \oplus K_{\lfloor n/2 \rfloor,A+i} \oplus H^{l-m}(r))$.

StepR3 : Sensor A stores $K_{i,A}$, $K_{i,(A+i) \bmod n}$, (i=1 to $\lfloor n/2 \rfloor$), and $H^l(r)$ into its memory.

StepR4 : KDS broadcasts m, $H^{l-m}(r)$ and the new joining sensor ID A to require the sensors in the network to update the key in common with the new joining one.

StepR5 : When a sensor B in the network receives the broadcast message, to ensure the authenticity of the message, sensor B checks whether $H^m(H^{l-m}(r))$ is equal to the stored $H^l(r)$ or not. If the equation holds, sensor B finds out which key of its pre-distributed keys is also held by the sensor A based on their IDs A and B. The process of finding out the common key CK_{A-B} is the same as the StepS2 of Section 3.2. After that, sensor B updates $CK_{A-B}=H(K_{1,B} \oplus K_{2,B} \oplus \ldots \oplus K_{\lfloor n/2 \rfloor,B} \oplus H^{l-m}(r))$.

For example, KDS assigns ID 2 to a new joining sensor, and **Fig. 6** illustrates the keys $\{K_{1,2}, K_{2,2}, K_{3,2}, K_{1,(2+1)}, K_{2,(2+2)}, K_{3,(2+3)}\}$ assigned to the new joining sensor after they are updated. When sensor 1 received the broadcast message $\{m, H^{l-m}(r), 2\}$ sent by the KDS, it first authenticates the message, and then finds out the common key $CK_{1\text{-}2}=K_{|1\text{-}2|,\max(1,2)}=K_{1,2}$. After that, sensor 1 updates $K_{1,2} = H(K_{1,1} \oplus K_{2,1} \oplus K_{3,1} \oplus H^{l-m}(r))$. The other sensors do the same process to update the key in common with the new joining sensor.

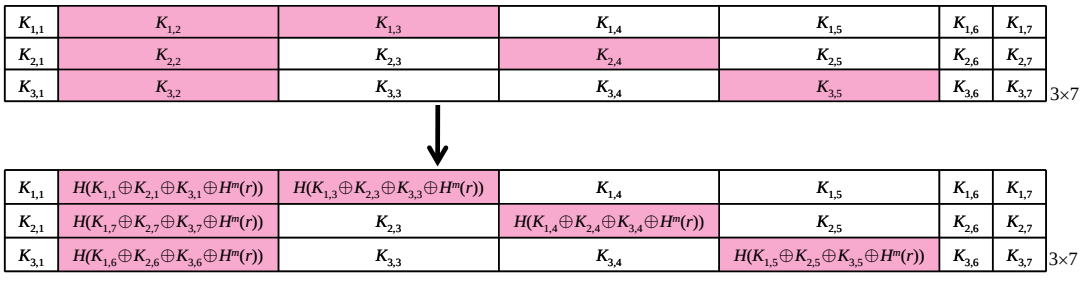

Fig. 6. An example of key refreshment (Joining Sensor's ID is 2)

3.4 Group Key Establishment Phase

When sensors desire to have secure group communication, they form a group at first, and then the KDS is responsible to generate a group key GK for those sensors. The group is denoted as S_G containing all the IDs of the sensors in the group. For each sensor j in S_G, the KDS constructs a unique line for it. The line passes through the two points (0, GK) and (Key_j, $H(Key_j$

$\|j$)), where $Key_j = (K_{1,j} \oplus K_{2,j} \ldots \oplus K_{\lfloor n/2 \rfloor, j}) \oplus r_G$, and r_G *is a random* number. In other words, each line can be expressed as an one-degree polynomial $f_j(x)=Key_j^{-1} \cdot (H(Key_j \| j)-GK) \cdot x+GK$. Note that the group key GK is stored at the constant term of each polynomial $f_j(x)$. GK is called as the secret of the polynomial. By the (2, 2) threshold secret sharing technique [18], the secret can be restored by combining two shadows of the polynomial. Here one point of the line corresponds to one shadow of the polynomial. KDS computes $f_j(j)$ and broadcasts r_G and $\{f_j(j), j \in S_G\}$. Each sensor j in S_G can derive the point $(Key_j, H(Key_j \| j))$ by itself, and get the other point $(j, f_j(j))$ from the broadcast message. The group key GK can be restored by each sensor after having the two points. **Fig. 7** illustrates the broadcast message at the group key establishment phase, and the process of establishing a group key is depicted as follows. It is assumed that the group consists of t sensors.

StepG1 : The KDS generates a group key GK and a random number r_G.

StepG2 : For each sensor j in S_G, the KDS constructs a line $f_j(x)$ through the two points $(0, GK)$ and $(Key_j, H(Key_j \| j))$, where $Key_j = (K_{1,j} \oplus K_{2,j} \ldots \oplus K_{\lfloor n/2 \rfloor, j}) \oplus r_G$. The line $f_j(x) = Key_j^{-1} \cdot (H(Key_j \| j)-GK) \cdot x+GK$. After that, the KDS computes $f_j(j)$.

StepG3 : The KDS broadcasts $\{r_G, \{f_j(j), j \in S_G\}, auth_G\}$, where $auth_G = H(r_G \| GK)$ is the message authenticator.

StepG4 : For each sensor j in S_G, after receiving the broadcast message, it gets the point $(j, f_j(j))$ by its ID and the message, and then it computes $Key_j = (K_{1,j} \oplus K_{2,j} \ldots \oplus K_{\lfloor n/2 \rfloor, j}) \oplus r_G$ to get the other point $(Key_j, H(Key_j \| j))$. After that, each sensor j can recover the group key GK by applying the equation $GK = f_j(0) = (Key_j \cdot y_j) - (j - H(Key_j \| j)) \cdot (Key_j - j)^{-1}$. Finally, sensor j computes $H(r_G \| GK)$ and then checks whether the $auth_G$ of the message is equal to the computed $H(r_G \| GK)$ to validate the integrity and authenticity of the group key.

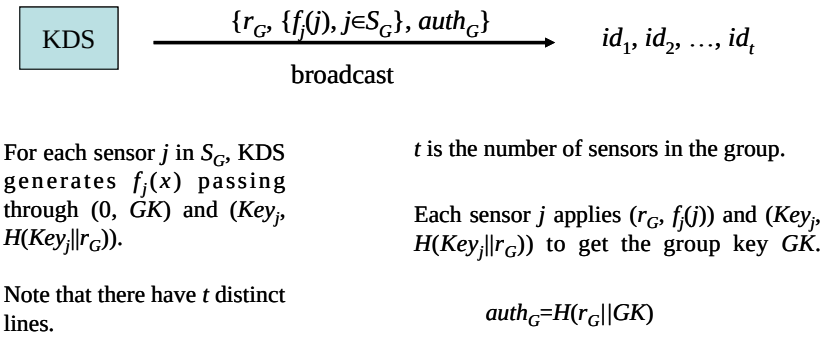

Fig. 7. Group key establishment phase

4. Security Analysis

In this section, we discuss the security properties of the shared key and the group key respectively. By our scheme, a shared key is generated by two sensors individually. In the

following, we will show that such a scheme not only achieves authenticity, consistency and flexibility properties, but also has the fully key connectivity and is able against the node capture attack. When secure group communication is required by more than two sensors, the KDS generates a group key and then each sensor by itself in the group is able to restore the group key based on the secret sharing technique. Mutual authentication between a sensor and the KDS is enforced while the group key is generated, and the backward secrecy and the forward secrecy are guaranteed.

4.1 Shared Key

4.1.1 Message Authenticity

At the shared key establishment phase of Section 3.2, the message authenticity is ensured by finding out a same common key CK to be used to derive a same shared key SK, and then the shared key is applied to generate the message authenticator $H(SK_{A\text{-}B}||r_A||r_B)$. As for the key refreshment phase, the hashing value $H^l(r)$ stored at sensors is used to validate the authenticity of the broadcast message sent by the KDS.

4.1.2 Consistency

The *consistency* property is that a same shared key can be derived when any two sensors needs it. In our scheme, a shared key can be derived by two sensors themselves without the KDS intervention. As mentioned above, our scheme requires that any two sensors must authenticate each other before deriving a shared key. Since the two sensors have the same random numbers and the common key, they drive a same shared key.

4.1.3 Flexbility

The *flexibility* property means that a sensor can leave or join the network at any time. Recall that a key in the matrix K is assigned to two sensors at most. That makes no key needed to be refreshed when a sensor leaves the network. As for a sensor joining the network, only one broadcast message is needed to notify sensors to update the keys in common with the new joining sensor.

4.1.4 Full Key Connectivity

The proposed quorum system assures that the intersection of any two subsets is non-empty. It means that any two sensors can find out a common key from their pre-distributed keys, a shared key is established based on the common key without the KDS intervention.

4.1.5 Against node capture attack

When a sensor was captured by an attacker, it is assumed that the captured sensor can be detected by some tools like the intrusion detection system and then the KDS is informed of the ID of the captured sensor [6]. After that, KDS announces the ID of the captured sensor. After receiving the ID of the captured sensor, the remaining sensors regard it as a leaving sensor. A sensor won't communicate with a leaving sensor. Thus, it is impossible for the attacker to have a same common key with any sensor even if the attacker spoofed some sensor's ID. This is because a common key is derived by two sensors separately, and the derived keys are the same only if the pre-distributed keys match the sensors' IDs.

Table 2 depicts the comparison of the works of [4][6][8] and our scheme.

Table 2. Security comparison

Security \ Scheme	KMSDSN [8]	EPKEM [4]	EKPSN [6]	Our scheme
Authenticity	No	No	No	Yes
Consistency	No	No	No	Yes
Flexibility	Yes	Yes	Yes	Yes
Fully key connectively	No	Yes	Yes	Yes
Against node capture attack	No	No	Yes	Yes

4.2 Group Key

4.2.1 Mutual authentication

In this paper, the mutual authentication in the group communication is referred to as a sensor can authenticate that a message sent by the KDS, and only a legal sensor can obtain the group key. At the group key establishment phase of section 3.4, each sensor in the group can generate the group key by itself after receiving the message broadcast by the KDS. Each sensor j authenticates the KDS by validating the message authenticator $Auth_G = H(r_G||GK)$. Since only the KDS and the sensor j know the whold column $K_{*,j}$ of the key matrix, only the legal sensor j is able to restore the correct group key GK. Although the KDS can not know which sensors obtain the group key, it indicated that the KDS authenticates the legal sensor implicitly.

4.2.2 Backward secrecy & Forward secrecy

Secure group communication indicates that the group data is encrypted/decrypted by a group key to provide message confidentiality. The backward secrecy means that any joining sensor can not access the pass group data, and the forward secrecy means that any leaving sensor can not access current or future goup data. In our scheme, whenever a sensor joins into or leaves from the group, the KDS is responsible to generate a group key randomly. Thus, it is impossible to derive a future or past group key from the current group key since all group keys are independently generated.

5. Performance Evaluation

In this section, we compare the performance of the proposed scheme with the works of [4][6][8]. **Table 3** gives the comparing results of the number of message sent by each sensor. During the shared key establishment process, the works of [4][6][8] and our scheme, each sensor only sends out one message to the other. However, the work of [8] requires that the message contains a list of pre-assigned keys to find a common key. The message size of [8] is longer than the others.When a sensor leaves from the network, the work of [8] requires that the KDS broadcasts a message containing a list of keys held by the leaving sensor. In our scheme, the KDS only needs to announce the leaving sensor's ID and the sensors in the network do not need to update any of their pre-distributed keys. As for a sensor joins into the network, the works of [4] and [6] regard it as system restart to execute the key pre-distribution phase. Our scheme requires the KDS to broadcast the ID of the joining sensor, and then each sensor updates only one key which is common with the new joining one.

The works of [4][6][8] do not discuss group key. This paper extends a group key establishment scheme to let sensors in a group can restore the group key without cooperating with other sensors. Only one message is broadcast by the KDS.

Table 3. Number of message sent by each sensor

		KMSDSN [8]	EPKEM [4]	EKPSN [6]	Our scheme
Pair-wise Key Establishment	Sensor	1	1	1	1
	KDS	0	0	0	0
Sensor Leaving	Sensor	0 or 1	0	0	0
	KDS	1	1	1	1
Sensor Joining	Sensor	0	*re*	*re*	0
	KDS	0	*re*	*re*	1
Group Key Establishment	Sensor	*NA*	*NA*	*NA*	0
	KDS	*NA*	*NA*	*NA*	1

NA: Not Avaliable
re: re-executeing the key pre-distribution phase

Both of the communication and the computation overhead for sensors is light since no message is required to be sent out and each group sensor only needs to execute two modular multiplications, one modular inverse, two hash operations and some XOR operations to restore the group key.

Table 4 gives the comparing results of the number of keys being maintained by the KDS and each sensor. In our scheme, the KDS maintains a $\lfloor n/2 \rfloor \times n$ key matrix K, and each sensor j stores $K_{i,j}$ and $K_{i,j+i}$ (for i=1 to $\lfloor n/2 \rfloor$). Our scheme provides full key connectivity without executing the path-key establishment procedure as the work of [8] did. Besides, our scheme can easily update keys by each sensor without re-assigning all sensors keys as the works of [4] and [6] did.

Table 4. Number of maintained keys

	KMSDSN [8]	EPKEM [4]	EKPSN [6]	Our scheme
Each Sensor	n_p+1	$2(n^{1/2})$	λ+3	n-1
key pools	2^{17}~2^{20}	n	λn	$\lfloor n/2 \rfloor \times n$

n_p: The number of pre-assigned keys [8]
n: Network size
λ: Security parameter [6]

6. Conclusion

We propose a quorum-based key pre-distribution scheme for wireless sensor networks. Based on the characteristic of a quorum system - the intersection of two subsets is non-empty. That makes our scheme to achieve fully key connectivity - any two sensors can find a common key if needed. The proposed quorum system guarantees that any key in the matrix K is held by two sensors at most. That makes no key needed to be refreshed when a sensor leaves the network. When a sensor joins the network, only a broadcast message is sent out by the KDS, and then sensors execute XOR and hash operations to update the key in common with the new joining

one, and only one key is updated for each sensor. Besides, a group key can be generated by each sensor in a group if secure group communication is required. The overhead for generating a group key is one broadcast message sent out by the KDS, and then each sensor in the group applies Lagrange interpolation polynomial to recover the group key by executing two modular multiplications, two hash operations, one modular multiplication inverse and some XOR operations. The proposed scheme has low message traffic and low computation overhead to make it appropriate for dynamic wireless sensor networks.

To save memory usage, the scheme claims that the pre-distribution keys are stored at a $\lfloor n/2 \rfloor \times n$ matrix, where n is odd. That makes each sensor to have n-1 keys after the key pre-distribution phase. If n is even, the number of sotred keys for each senosr would be n, one more key than n is odd. However, the scheme works and has the same security no matter how n is odd or even. The difference is that, when n is even, sensor A holds two keys $K_{n/2, A}$ and $K_{n/2, (A+(n/2)) \bmod n}$ which are also held by the sensor $(A+(n/2)) \bmod n$. However, only one common key $CK_{A-(A+n/2)}=K_{n/2, \max(A, (A+n/2) \bmod n)}$ can be derived by sensor A and sensor $(A+(n/2)) \bmod n$. It means that the other key is useless. Thus, that n is odd sensors will not wast memory storage to maintain unused keys.

References

[1] David W. Carman, Peter S. Kruus and Brian J. Matt, "Constraints and approaches for distributed sensor network security," *Network Associates Inc*, 2000. Article (CrossRef Link)

[2] Haowen Chan, Adrian Perrig and Dawn Song, "Random key predistribution schemes for sensor networks," in *Proc. of the IEEE Symposium on Security and Privacy*, pp.197-213, May.2003. Article (CrossRef Link).

[3] Ni Chen, Jian-Bo Yao and Guang-Jun Wen, "An improved matrix key pre-distribution scheme for wireless sensor networks," in *Proc. of the Int. Conf. on Embedded Software and Systems*, pp.40-45, Jul.2008. Article (CrossRef Link).

[4] Yi Cheng and Dharma P. Agrawal, "Efficient pairwise key establishment and management in static wireless sensor networks," in *Proc. of IEEE Int. Conf. on Mobile Ad hoc and Sensor Systems*, Nov.2005. Article (CrossRef Link).

[5] Yi Cheng and Dharma P. Agrawal, "Improved pairwise key establishment for wireless sensor networks," in *Proc. of IEEE Int. Conf. on Wireless and Mobile Computing, Networking and Communications*, pp.442-448, Jun. 2006. Article (CrossRef Link).

[6] Hung-Yu Chien, Rung-Ching Chen and Annie Shen, "Efficient key pre-distribution for sensor nodes with strong connectivity and low storage space," in *Proc. of 22th Int. Conf. on Advanced Information Networking and Applications*, pp.327-333, Mar.2008. Article (CrossRef Link).

[7] Wenliang Du, Jing Deng, Yunghsiang S. Han, Pramod K. Varshney, Jonathan Katz and Aram Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security*, vol.8, no.2, May.2005. Article (CrossRef Link).

[8] Laurent Eschenauer and Virgil D. Gligor, "A key-management scheme for distributed sensor networks," *In Proc. of the 9th ACM conf. on Computer and communications security*, pp.41-47, Nov.2002. Article (CrossRef Link).

[9] Donggang Liu, Peng Ning and Rongfang Li, "Establishing pairwise keys in distributed sensor networks," *ACM Transactions on Information and System Security*, vol.8, no.1, Feb.2005. Article (CrossRef Link).

[10] Kui Ren, Kai Zeng and Wenjing Lou, "A new approach for random key pre-distribution in large-scale wireless sensor networks," *Journal of Wireless Communications & Mobile Computing - Wireless Network security*, vol.6, no.3, May.2006. Article (CrossRef Link).

[11] Eric Ke Wang and Yunming Ye, "An efficient and secure key establishment scheme for wireless sensor network," *in Proc. of the Third Int. Symposium on Intelligent Information Technology and Security Informatics*, pp.511-516, Apr.2010. Article (CrossRef Link).

[12] Li Xu and Jinbo Shen, "A novel key pre-distribution scheme using one-way hash chain and bivariate polynomial for wireless sensor networks," in *Proc. of the 3rd Int. Conf. on Anti-counterfeiting, Security, and Identification in Communication*, pp.575-580, Aug.2009. Article (CrossRef Link).
[13] Rolf Blom, "An optimal class of symmetric key generation systems," in *Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques*, pp. 335-338, 1985. Article (CrossRef Link)
[14] S.D. Lang and L.J. Mao, "A torus quorum protocol for distributed mutual exclusion," in *Proc. of the 10th Int. Conf. on Parallel and Distributed Computing and Systems*, 1998. Article (CrossRef Link)
[15] Jehn-Ruey Jiang, Shing-Tsaan Huang and Yu-Chen Kuo, "Cohorts structures for fault-tolerant k entries to a critical section," *IEEE Transactions on Computers*, vol.46 no.2, Feb.1997. Article (CrossRef Link).
[16] Mie Toyomura, Sayaka Kamei and Hirotsugu Kakugawa, "A quorum-based distributed algorithm for group mutual exclusion," in *Proc. of the Fourth Int. Conf. on Parallel and Distributed Computing, Applications and Technologies*, pp.742-746, Aug.2003. Article (CrossRef Link).
[17] David Peleg and Avishai Wool, "The availability of crumbling wall quorum systems," *Journal of Discrete Applied Mathematics*, Vol. 74, No. 1, April 4, 1997. Article (CrossRef Link).
[18] Adi Shamir, "How to share a secret," Communications of the ACM, vol.22, no.11. pp.612-613, Nov.1979. Article (CrossRef Link).

Lih-Chyau Wuu received her B.S. degree in the department of information engineering from National Taiwan University, Taipei, Taiwan, in 1982, and her Ph.D. degree in the department of computer science from National Tsing Hua University, Hsinchu, Taiwan in 1994. She is currently a Professor in the department of computer science and information engineering, National Yunlin University of Science & Technology, Touliu, Taiwan. Her research interests include IP switches /routing, multicast routing, network security and distributed self-stabilizating systems.

Chi-Hsiang Hung received his B.S. and M.S. degrees in the department of electronic engineering from National Yunlin University of Science & Technology (Touliu, Taiwan), in 2003 and 2006, respectively. He is currently a Ph. D. candidate in graduate school of engineering science and technology-doctoral program, National Yunlin University of Science & Technology (Touliu, Taiwan). His research interests include network security, information security.

Chia-Ming Chang received his B.S. degrees in the department of computer science and information engineering from National Formosa University (Huwei, Taiwan) in 2008 and his M.S. degrees in the department of computer science and information engineering from National Yunlin University of Science & Technology (Touliu, Taiwan) in 2010.