Cover-Free Family based Efficient Group Key Management Strategy in Wireless Sensor Network

Li Xu
Key Lab of Network Security and Cryptology, Fujian Normal University, Fuzhou, China
Email: xuli@fjnu.edu.cn

Jianwei Chen and Xiaoding Wang
Key Lab of Network Security and Cryptology, Fujian Normal University, Fuzhou, China
Email: {cjwin, sjky}@fjnu.edu.cn

***Abstract*— Secure group key distribution and efficient re-keying is one of the most challenging security issues in sensor networks at present. In this paper, Latin square is firstly used to construct orthogonal arrays in order to obtain t-packing designs quickly. Based on cover-free family properties, t-packing designs are adopted in key pre-distribution phase. Then based secure key-shared method, the pre-deployed keys are used for implementing secure channels between members for group key distribution. The efficient updating the pre-deployed keys scheme is used to deal with the variety of network. The new strategy improves the collusion-resilience of the networks using the cover-free family properties, and enhances the key-sharing connectivity of nodes with which makes key management more efficiently. This paper also presents in depth theory and data analysis of the new strategy in term of network security and connectivity**

***Index Terms*—Wireless Sensor Network, Cover-Free Family, Key pre-distribution, T-packing design**

I. INTRODUCTION

Wireless sensor network are ad-hoc networks comprised mainly of small sensor nodes with limited resources (low power, low bandwidth, and low computational and storage capabilities) and one or more base stations (BSs), which are much more powerful nodes that connect the sensor nodes to the rest of the world. Wireless sensor network are being deployed for a wide variety of application, including military sensing and tracking, environment monitoring, patient monitoring and tracking, smart environments, etc.

For deploying these applications, it is necessary to provide support for secure multicast and broadcast communication. The most efficient approach for achieving confidential group communication is to use a symmetric group key that is shared by all the nodes for data encryption. This approach however introduces the problem of group re-keying, i.e., the group key must be updated and redistributed to all the remaining nodes in a secure, reliable and timely fashion when group membership changes. Therefore this problem requires key management systems that provide support for dynamic properties.

A. Related Work

In general, key management systems are of three types: namely key distribution, key agreement and key pre-distribution. The traditional Internet style key distribution protocols, for example Kerberos [2] or adapted LKH schemes [3], are infeasible for sensor networks because of their exclusive properties. These include communication range limitations, node and network dynamics and unknown network topology prior to deployment. On the other hand, contributory key agreement protocols [4,5,6,7], in which each node contributes an input to establish a common secret through successive pairwise message exchanges among the nodes in a secure manner by using the 2-party Diffie-Hellman exchange, are not practical to sensor networks either. They are not robust to changing topology or intermittent links both of which commonly occurred in a sensor network. In order to successfully establish a key, these protocols strictly require the underlying networks to either support broadcasting or have a relatively time-invariant topology of certain forms. The key agreement approach is not scalable due to the need of frequent interactive re-keying despite key freshness. Therefore naturally that we are interested in key pre-distribution schemes (or KPS), where a set of secret keys is install in each node before the sensor nodes are deployed.

A number of recent works demonstrate that the key pre-distribution scheme (KPS) offers practical and efficient solutions to the key management problem. In KPS, each node receives a subset of keys from a key pool before deployment. Any two nodes are able to find or compute non-interactively common keys within their respective subsets and they can use that key as its shared secrets to initiate communication.

There are some successful key pre-distribution schemes. Eschenauer and Gligor [8] proposed a random key pre-distribution scheme. Each sensor node receives a subset of random keys from the pool before deployment. Chan, Perrig, and Song [9] proposed a q-composite random key pre-distribution scheme, which increases the security of key setup so that an attacker has to compromise many more nodes to achieve a high probability of compromising communication. Recently, Du et al. [22] proposed another key pre-distribution scheme in which substantially improves the resilience of the network compared to other schemes. Chan [10] proposed a fully distributed key pre-distribution scheme (DKPS) with no trusted authority for sensor networks. The DKPS is based on the precondition that the key sets distributed to the network nodes can form a cover-free family. Wu and Wei [11] found that the precondition was falsely deduced. They claim that the probabilistic method (Chan used this method) cannot yield CFF practical for key distribution.

GKMPAN is an efficient group re-keying scheme for secure multicast in ad-hoc networks proposed by Zhu [12][19]. GKMPAN also uses the probabilistic key pre-distribution technique as the underlying means to establish secure channels between nodes. However, compared the previous schemes, GKMPAN uses the pre-deployed keys only as key encryption keys (KEKs) for securely distributing a group key to the nodes in the network while using the group key for securing group data communications. Thus, GKMPAN incurs much smaller communication and computational overhead among group communication. GKMPAN also includes an efficient mechanism to update the pre-deployed keys of nodes. So the performance comparison in Section 5 is closet to the one between our schemes and GKMPAN.

Based on the analysis of different strategies above, A good key management scheme based on KPS for wireless sensor networks must be considered.

Connectivity: A network node should be able to securely communicate to its local neighbors. Here a local neighbor means a network node physically located within transmission range.

Resilience of the network: Even a quite amount of nodes are compromised by adversary, the communications between other nodes should be still secure.

Small key size: Since the limited resource of a node, key storage should be small. Therefore the number of keys distributed to a node should be small.

Number of nodes: How much nodes at maximum the scheme can support should be considered.

B. Our Contribution

In this paper, we propose a new t-Packing Design based Group Re-keying Scheme (PDGRS) for sensor networks. PDGRS builds on t-packing designs to pre-distribute node key-chains, and these keys are used for group re-keying. For this purpose, Latin squares are used to construct orthogonal arrays for quickly obtaining t-packing designs. This method makes the scheme mathematical model achieve the cover-free family (CFF) properties [15], which improves the collusion-resilience of the networks. Moreover, updating the pre-deployed keys further prevents the more compromised and revoked nodes from launching a collusive attack. The analysis in detail can be seen in the subsection 4.2. Meanwhile, PDGRS enhances the key-sharing connectivity of nodes with which makes keys distribution more efficient. We describe it in the subsection 4.3. Compared with other existing schemes, the total analysis shows that not only the key-sharing connectivity but also the collusion-resilience of the networks improves as the number of keys in a node increases . This character takes some advantage over other previous strategy in involved field.

The rest of this paper is organized as follows. In Section 2, we give the preliminaries and network model. Then, we present the details of the new scheme in Section 3. We discuss and analyze some security, efficient performance and key-sharing connectivity of the new strategy in Section 4. Finally, the results give out in Section 5.

II. PRELIMINARIES

A. Cover-free Families Mathematical Mode

Cover-free families were first studied in terms of superimposed binary codes by Kautz and Singleton [13] in 1964. These codes are related to retrieval files, Data communication and magnetic memories. Since then, cover-free families have been discussed in several equivalent formulations in subjects such as information theory, combination and group testing by researchers. Ling and Wang give an overview [21] of several interesting applications topics in secure networks and distributed including the key distribution patterns that is the topic in this paper.

A set system is a pair (X, F), where X is a set of points and F is a set of blocks of X. The classical definitions of cover-free families [14] can be written as follows.

Definition1. A set system (X, F) is called a r cover-free family (or r-CFF) provided that, for any r blocks $A_1, A_2, \Lambda, A_r \in F$ and any other block $B_0 \in F$, we have

$$B_0 \not\subseteq \bigcup_{j=1}^{r} A_j \tag{1}$$

Definition 2. A set system (X, F) is called a (r; d) cover-free family (or (r; d)-CFF) provided that, for any block $B_0 \in F$ and any other r blocks $A_1, A_2, \Lambda, A_r \in F$, we have

$$\left| B_0 \setminus \bigcup_{j=1}^{r} A_j \right| > d \tag{2}$$

The definition 2 states that the union of any r blocks contains at least d points being not in it. Combinatorial designs can be used to constructed *r-CFF*. First we give the definition of a *t*-packing design as follows, and then other related definitions.

Definition 3. A t-(v, k, λ) packing design is a set

system (X, F), where |X| = v, |B| = k for every $B \in F$, and every t-subset of X occurs in at most λ blocks in F.

Definition 4. A $k \times v^t$ array A with entries from V is an orthogonal array with v levels and strength t (for some t in the range $o \le t \le k$) if every $t \times v^t$ sub-array of A contains each t-tuple based on V exactly once (we assume the index λ=1) as a column. We denote such an array by $OA(t,k,v)$.

Definition 5. A Latin square of order n is an n by n array containing symbols from some alphabet of size n, arranged so that each symbol appears exactly once in each row and exactly once in each column.

B. Network Assumptions and Notations

We assume a wireless sensor network with N nodes. Network nodes communicate with each other and require pair-wise keys to secure their communication for group re-keying. Each node has a key-chain of k keys from which a key pre-distribution phase is selected based on packing designs before the deployment. After that any two neighbors nodes find the common keys between their key-chains using cryptography homomorphism with secure shared key discovery (SSD) [10] or Private Matching [20] based on the cryptology system and the security request in the network, and these keys are used to secure their communication. When a node joins or a member node leaves a group, the group key must be updated to enforce forward or backward secrecy. In addition, the pre-deployed keys need to be renewed. The notations in Table 1 will appear in the rest of this paper.

TABLE I.
NOTATIONS

N	Number of nodes
n	Number of neighbor nodes
P	The key pool
p	A key in the key pool
q	A prime or a prime power
$E_k(msg)$	The encryption of message *msg* with key k
$\{f_i\}$	A family of pseudo-random functions [16]
R_u	The key-chain of node u
m	Number of keys in a key-chain

III. T-PACKING DESIGN BASED GROUP RE-KEYING STRATEGY (PDGRS)

A. Overview of the Strategy

The strategy consists of the following phases.

Initial Setup Phase: The group controller (GC) selects parameters used in the scheme.

Key Pre-distribution Phase: Prior to the deployment of the sensor network, all nodes obtain a distinct subset of keys from the GC, based on packing designs.

Shared-key Discovery Phase: Nodes perform a protocol to discover their shared keys with their neighbors. Two nodes with shared keys are assumed securely connected. Next these keys are used as KEKs for delivering group keys.

Path-key establishment Phase: assigns a path-key to selected pairs of sensor nodes in wireless communication range that do not share a key but are connected by two or more links at the end of the shared-key discovery phase.

Key Update: After a node receives and verifies the group key K, it updates its own pre-deployed keys based on K.

B. Setup Phase

The key pool *P*, parameters *q* and *m* are both chosen by the GC. The choice of these parameters will determine the security level, the number of keys that a node has to store and communication efficiency of group key setup.

The number of keys the key pool *P* has is *q*. It is one-to-one mapping between the key pool *P* and the finite field *GF*(*q*) , that is , $P = \{p_i \mid i \in GF(q)\}$.

C. Key Pre-distribution Phase

Step1. Construct mutually orthogonal Latin squares of order *n* according to the following theorem.

Theorem 1. *Select a primitive element a from a finite field GF(n), then*

$$B_{i+1} = \begin{pmatrix} 0 & 1 & a & \Lambda & a^{n-2} \\ a^i & 1+a^i & a+a^i & \Lambda & a^{n-2}+a^i \\ a^{i+1} & 1+a^{i+1} & a+a^{i+1} & \Lambda & a^{n-2}+a^{i+1} \\ \Lambda & \Lambda & \Lambda & \Lambda & \Lambda \\ a^{i+n-2} & 1+a^{i+n-2} & a+a^{i+n-2} & \Lambda & a^{n-2}+a^{i+n-2} \end{pmatrix}$$

For $i = 0,1,\Lambda\ ,n-2$, is a complete set of orthogonal Latin squares of order n.

Step 2. Over the complete set of orthogonal Latin squares of order *n*, an orthogonal array $OA(t,k,v)$ can be constructed by the way described in [15].

Step 3. In this step, suppose $\{s_1, s_2, \mathrm{L}\ , s_k\}$ is a column in $OA(t,k,v)$. Define a block as $\{(0,s_1),(1,s_2),\mathrm{L}\ ,(k-1,s_k)\}$ accordingly. In this way, we can obtain a $t-(ks,k,1)$ packing design from the $OA(t,k,v)$.

After the packing design has been constructed, each node is loaded with the following information:

Information1. Each node u is loaded with R_u, which contains keys computed from the equation (3), and these keys are used as KEKs. Specifically, for each node, the GC chooses a block $B = \{(j,i) \mid j = 0,1,2,\Lambda\ q; i \in GF(q)\}$. Next the block is used to calculate the corresponding keys according to the equation (3) .

$$k_j = H(j, p_i), (j,i) \in B \qquad (3)$$

Information2. Each node is loaded with the initial group key k_g .

Example1. We illustrate the proposed phase using an example below, involving the construction of a packing design.

Step1. Assume that $q=5$, $GF(5)=\{0,1,2,3,4\}$. And the GC generates a key pool $P=\{p_0,p_1,p_2,p_3,p_4\}$.

Step2. Construct a complete set of 4 mutually orthogonal Latin squares of order 5 as follows.

$$\begin{pmatrix}0&1&2&4&3\\1&2&3&0&4\\2&3&4&1&0\\4&0&1&3&2\\3&4&0&2&1\end{pmatrix}\begin{pmatrix}0&1&2&4&3\\2&3&4&1&0\\4&0&1&3&2\\3&4&0&2&1\\1&2&3&0&4\end{pmatrix}$$

$$\begin{pmatrix}0&1&2&4&3\\4&0&1&3&2\\3&4&0&2&1\\1&2&3&0&4\\2&3&4&1&0\end{pmatrix}\begin{pmatrix}0&1&2&4&3\\3&4&0&2&1\\1&2&3&0&4\\2&3&4&1&0\\4&0&1&3&2\end{pmatrix}$$

Step3. We construct 6×25 $OA(2,6,5)$ using above mutually orthogonal Latin squares. Note that how many Latin squares we apply will determine the number of elements that a block has, that is, the number of keys that a node has. Assume here that we use all Latin squares. As a result, $OA(2,6,5)$ is obtained as follows.

$$\begin{pmatrix}0&0&0&0&0&1&1&1&1&1&2&2&2&2&2&3&3&3&3&3&4&4&4&4&4\\0&1&2&3&4&0&1&2&3&4&0&1&2&3&4&0&1&2&3&4&0&1&2&3&4\\0&1&2&4&3&1&2&3&0&4&2&3&4&1&0&4&0&1&3&2&3&4&0&2&1\\0&1&2&4&3&2&3&4&1&0&4&0&1&3&2&3&4&0&2&1&1&2&3&0&4\\0&1&2&4&3&4&0&1&3&2&3&4&0&2&1&1&2&3&0&4&2&3&4&1&0\\0&1&2&4&3&3&4&0&2&1&1&2&3&0&4&2&3&4&1&0&4&0&1&3&2\end{pmatrix}$$

Step4. Finally the following $2-(30,6,1)$ packing design is derived .

$$\begin{pmatrix}(0,0)&(0,0)&(0,0)&(0,0)&(0,0)&(0,1)&(0,1)&(0,1)&(0,1)&(0,1)&L\\(1,0)&(1,1)&(1,2)&(1,3)&(1,4)&(1,0)&(1,1)&(1,2)&(1,3)&(1,4)&L\\(2,0)&(2,1)&(2,2)&(2,4)&(2,3)&(2,1)&(2,2)&(2,3)&(2,0)&(2,4)&L\\(3,0)&(3,1)&(3,2)&(3,4)&(3,3)&(3,2)&(3,3)&(3,4)&(3,1)&(3,0)&L\\(4,0)&(4,1)&(4,2)&(4,4)&(4,3)&(4,4)&(4,0)&(4,1)&(4,3)&(4,2)&L\\(5,0)&(5,1)&(5,2)&(5,4)&(5,3)&(5,3)&(5,4)&(5,0)&(5,2)&(5,1)&L\end{pmatrix}$$

After the packing design is completed, the GC selects each node's, say u, block upon the input of its id. Suppose $B_u=\{(0,0),(1,4),(2,3),(3,3),(4,3),(5,3)\}$. And the GC calculates its corresponding key-chain distributed to node u as equation (4) according to the equation (3).

$$R_u=\{H(0,p_0),H(1,p_4),H(2,p_3),H(3,p_3),H(4,p_3),H(5,p_3)\} \quad (4)$$

D. Direct or Indirect Key path Discovery

After the key pre-distribution phase is completed, each node is deployed in different places to set up the network. Any two neighbors node have or have not common key, so there will be a method to find if they have shard-key. If two nodes are out of communication range, they must to find a key path making use of other nodes. Discussion on how to tradeoff between security, connectivity and storage cost is also presented at the end of this sub-section.

If neighbor nodes u and w need to set up a secure communication link they need to check whether they have a common key in their pre-distributed sets of keys. This phase was named as neighbor node's shared-key discovery phase (or SKD). A simple SKD method is to allow the two sensors to communicated with each other their list of key-Ids in plaintext like in reference [10][5][17]. They suppose the key pre-distribution algorithm is public and deterministic. But this kind method request SKD is finished short after the key pre-distribution phase and there is no adversary in the local communication range.

For example, after the deployment, u and w become neighbor nodes after the deployment, with which are respectively assigned key-chains as follows.

$$R_u=\{u_1,u_2,u_3,\Lambda\ u_k\}$$
$$R_v=\{v_1,v_2,v_3,\Lambda\ v_k\} \quad (5)$$

Based on the security request of the network and the application, we give out three optional methods to finish SKD as follow.

Methods 1:

In our preliminary work in the proceeding of INFORSCALE 2007 conference, we use SSD scheme [10], which uses privacy homomorphism to find common keys between neighbor nodes R_u and R_w. The SSD scheme allows two nodes to find out common keys in their key-chains, but not to leak out to the other side any information of the keys outside the common intersection of the two key-chains.

Method 2:

Note that, it is possible that the same key maybe shared by more than a pair of sensor nodes. Another probable situation is how to find out the more common key between sensor nodes at one time if they are had. It is useful when the network need more higher security request or key connectivity. To solve this problem, we propose the Identity based private matching strategy (IBPM) [20]. With the help of group controller (GC), Using bilinear pairing, IBPM include two methods: simple IBPM and IBPM with DOP (Data Ownership Parameters). They can prevent the guessing attack and address spoofing. The more detail can be seen from reference paper.

Method 3:

The Zp is a finite field, where p is a prime number and the discrete logarithm problem is hard in Zp. The g is a generator element of Fq. H is a hash function, H: Zp×Zp. U and V picks Sa and Sb∈Zp^* . Compute g_a and g_b as equation (6), then public g_a and g_b.

$$g_a=g^{s^a} \bmod p \quad g_b=g^{s^b} \bmod p \quad (6)$$

Node u and v compute the Hash value based on the public and private information as the equation (7). The advantage of method 3 is it can efficiently get more than one common keys.

$$H(u_1,g^{s_as_b}),H(u_2,g^{s_as_b}),\Lambda\ H(k_k,g^{s_as_b}),$$

$$H(v_1, g^{s_b s_a}), H(v_2, g^{s_b s_a}), \Lambda\ H(v_k, g^{s_a s_b}), \quad (7)$$

For the convenience of comparing with the GKMPAN strategy, we suppose that the shared key procedure is short after the key pre-distribution procedure and no node has been captured. So the simulation analysis in section 4 uses SSD scheme.

The path-key establishment phase assigns a path-key to selected pairs of sensor nodes in wireless communication range that do not share a key but connected by two or more links at the end of the shared-key discovery phase.

It is known that to increase the key-sharing connectivity and enhance the security, it is necessary to increase *m*, the number of keys a node stores. However, from the viewpoint of storage, a smaller *m* is more desirable. Due to these conflicting requirements, the parameter *m* should be selected based on the application under consideration. For example, in the case when the storage of a node seems to be too restrictive, we can increase the key-sharing connectivity by the way described in the section 4 under the precondition that the security requirements are ensured.

Indeed, PDGRS scheme use key pool, however GC is not directly divided the keys in key pool to nodes, but use orthogonal Latin squares function to construct the more keys. The method can lighten the GC's management load, facilitate key renew. On the other hands, from the analysis in section 4, however, we see that the actual local connectivity depends on the amount of space available on a node for storing keys, therefore, when the node resource is limited, we will improve the P_c by directly increasing the node degree *d*. PDGRS can use the following two ways to increase *d*.

The first way is that a node *u* can use its neighbors which have shared keys with *u*, to establish a secure channel with other nodes in *u*'s one-hop communication range. We take node *a* as an example. In node *a*'s one-hop communication range, node *b* has common keys with node *a* and node *c* respectively, but no common key exists between node *a* and node *c*. In this case, if node *a* want to establish shared keys with node *c*, It can ask node *b* to act as a proxy. Suppose node *a* shares a key k_{ab} with node *b*, node c shares a key k_{bc} with node *b*. To forward a key *k* to node *c*, the following steps are taken.

$$a \to b : E_{k_{ab}}(k), \quad b \to c : E_{k_{bc}}(k) \quad (8)$$

The second way is to use two-hop neighbors. A two-hop neighbor of node *u* is a node that can be reached via one of *u*'s one-hop neighbors. To send a message to a two-hop neighbor, *u* needs to ask its direct neighbor to forward the message. Node *b* has common keys with node *a* and node *c* respectively. Node *c* is similar to the case above, except that node *c* is out of node *a*'s one-hop but in two-hop communication range. Therefore node *a* asks node *b* to act as a proxy, not only to establish a secure channel with node *c* but also to forward messages to node *c*. For node *d*, it is also out of node *a*'s one-hop communication range, but has common keys with node *a*. So, in this case node *b* only needs to forward messages. Suppose node *a* shares a key k_{ad} with node *d*. To forward a message *msg* to node *d*, the following steps are taken.

$$a \to b : E_{k_{ad}}(msg), \quad b \to d : E_{k_{ad}}(msg) \quad (9)$$

E. Re-keying Phase

Our scheme does not require a key pre-distribution phase for every instance of network formation. In fact, there is no limit on how many times these pre-distributed keys can be used securely because our re-keying scheme updates these keys securely after every group re-keying.

Node Join: In this section, without losing generality, we suppose a new node u wants to join an existing group. For example, the GC may introduce new nodes into the system to compensate for revoked nodes. To enforce forward secrecy, the following steps will be adopted.

Step1. The GC generates a new group key $k_g^{'}$, and broadcasts the message $E_{k_g}(k_g^{'})$ to the network.

Step2. Every node, say *v*, updates every key k_i in R_v as $k_i^{'} = f_{k_i}(0)$. We denote the updated set of keys as $R_v^{'}$.

Step3. After the key update operations, every node erases the old group key k_g.

Step4. Finally, the GC determines *u*'s key set R_u based on its node id. Then it loads node *u* with current version of R_u and the current group key over a secure channel. Such a confidential and authentic channel can be established if user physically goes to the GC or the keys can be protected by a simple blinding technique [16].

Node Revocation: The leaving action may happen voluntarily in addition when a compromised node is detected and expelled from a group. Either way, the keys must be updated to enforce backward secrecy. Let u be the node to be revoked.

Step1. The GC determines *l* keys $\{k_1, k_2, \Lambda, k_l\}$, which are the non-compromised keys that are possessed by the remaining nodes in the network, and these keys are used as KEKs. The GC then generates a new group key $k_g^{'}$. Then it broadcasts a node revocation message as equation (10) to the network.

$$\mathrm{GC} \to *: ID_u,$$
$$\{E_{k_1}(k_g^{'}), E_{k_2}(k_g^{'}), \Lambda, E_{k_l}(k_g^{'})\}, \quad f_{k_g^{'}}(0) \quad (10)$$

Step3. The nodes that possess one of the *l* keys $\{k_1, k_2, \Lambda, k_l\}$ can compute the new group key $k_g^{'}$ independently. Otherwise, they can obtain it over the shared-keys with their neighbors. Node *u* will not receive $k_g^{'}$ even though it can impersonate a non-revoked node *v* by claiming node *v*'s id, because none of the keys in R_u are used. And node u also can not derive $k_g^{'}$ from its neighbors, since the node revocation message involves its node id.

Step4. After every node receives the new group key $k_g^{'}$, it verifies the correctness of $k_g^{'}$ by checking if

$f_{k_g'}(0)$ equals to that in the node revocation message. If equals, every node, say v, updates every key k_i in R_v as $k_i' = f_{k_i}(0)$. We denote the updated set of keys as R_v'.

Step5. After the key update operations, every node erases the old group key k_g.

In step 1, the l keys chosen by the GC can be the non-compromised keys that are possessed by the maximum number of nearby remaining nodes of the GC in the network. When a node possesses none of the l keys, it can obtain the group key over the shared-keys with its neighbors. As long as the key-sharing connectivity of nodes is high, the group key will be efficiently distributed to the remaining nodes in the network.

Ⅳ. Performance Analysis

In this section, we will compare the property of our new strategy with some of current study and analyze the security and the key-sharing connectivity.

A. Efficiency Analysis

Now we compare the properties of our scheme to that of Eschemauer et al.'s random key pre-distribution scheme (RKPS), Chan's distributed key pre-distribution scheme (DKPS) and Zhu et al.'s efficient and scalable group re-keying protocol (GKMPAN). Note that Eschenauer et al.'s scheme is the first one that uses the results from random graph theory and probabilistic method to manage keys in sensor network. Chan proposed a fully distributed key pre-distribution scheme with no trusted authority. And GKMPAN uses pre-deployed symmetric keys for implementing secure channels between members for group key distribution, which has some attractive properties.

TABLE 2.
Comparison of Properties

	RKPS	DKPS	GKMPAN	PDGRS
CC	yes	no	yes	yes
CFF	no	yes	no	yes
UPDK	no	no	yes	yes
EFBS	no	no	yes	yes
LKSL	no	no	no	yes

The common feature among the PDGRS strategy and the other three schemes is that the group formation is dynamic. User can join or leave the group (or be revoked from the group) at any time. However, We also note that these schemes have different properties. Table 2 summaries the properties that four different proposals have as to whether they require Centralized Control (CC), CFF properties, whether to updating the pre-distribution key (UPDK), whether to enforce forward and backward secrecy (EFBS), and whether to consider Lighten the Key Storage Load (LKSL). So we can found that our PDGRS have most good features.

B. Security Analysis

Except for forward and backward secrecy, the security of our group re-keying scheme is mainly twofold.

Theorem 2. *If there is an* $OA(t,k,v)$, *then there is a t —(kv, k, 1) packing design that contains* v^t *blocks.*

Proof. Supposes that there is an $OA(t,k,v)$ with entries from the set $\{0,1,\Lambda,v-1\}$. Defines $X=\{(x,y)\mid 0\le x\le k-1, 0\le y\le v-1\}$. For every column $(y_0,y_1,\Lambda,y_{k-1})$ in the orthogonal array, define a block $B=\{(0,y_0),(1,y_1),\Lambda,(k-1,y_{k-1})\}$. Let F consist of the v^t blocks thus constructed. It is easy to check that (X, F) is a $t-(kv,k,1)$ packing design.

A t-packing design is an r-CFF for certain value of r. We obtain the following construction.

Theorem 3. If there exists a t-(v, k, 1) packing design having b blocks, then there exists a (r; d)-CFF(v, b), where $r=\lfloor (k-d-1)/(t-1)\rfloor$.

In PDGRS, q is a prime or a prime power, and there exists a complete set of (q-1) mutually orthogonal Latin squares. Using definition 1 and the above lemmas, we can easily obtain the following result.

Corollary 1: For any prime power q and any integer t < q, then there exists an OA(t, q +1, q), such that a $t-(kq,k,1)$ *packing design with* q^t *blocks exists, so exists a* $(\left\lfloor \frac{k-d-1}{t-1}\right\rfloor, d)-CFF(qk,q^t)$, *where* $k\le q+1$.

Given $k=q+1$, we have the following.

$$(\left\lfloor \frac{q-d}{t-1}\right\rfloor, d)-CFF(q^2+q,q^t) \qquad (11)$$

Corollary 2: In the scheme PDGRS, when the number of colluding nodes is less than r, other secrete keys used by any other nodes can not be completely covered.

For example, we choose q=113, d=2 and the number of keys stored in a node m is 114, then the result r=111 is obtained. That is, at least two keys of any other legitimate nodes are secure, when the number of simultaneously colluding nodes is less than 111.

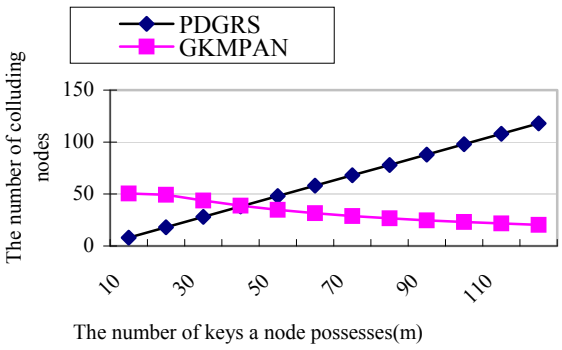

Figure 1. Comparison of the number of colluding nodes

In Fig.1 we compare the number of colluding nodes (denoted as *w*) that PDGRS and GKMPAN [12] can tolerate by varying the number of keys in a node. We can observe that the number of colluding nodes PDGRS resists increases with *m*, but GKMPAN inverses. In PDGRS, *w* and *m* are in direct proportion basically. While in GKMPAN, for a fixed probability 0.01% that a node is covered, the number of colluding nodes the scheme resists decreases with *m*. For example, for a group size of 10,000, when *m*=120, the coalition of only 20 nodes can lead to have keys to cover a legitimate node. Note other schemes [7, 8] have a similar result like GKMPAN.

Updating pre-deployed keys: To further improve the resilience, our scheme also updates the pre-deployed keys as GKMPAN. It is critical to prevent the more compromised and revoked nodes from launching a collusive attack in which they pool their keys together with the goal of jeopardizing other legitimate nodes. Without key updating, both the performance and security of the system will degrade greatly number of compromised nodes. That is, we only need to guarantee that the number of compromised or revoked nodes between two key refreshments is less than the threshold *r*, because the status of the system is reinstated to its original setting after every re-keying. Consequently, the security of our scheme can be strengthened largely.

C. Key-sharing Connectivity Analysis

As we have just shown, to make it possible for any node to be able find shared keys with its neighbors to secure group communication, the key sharing graph needs to be connected. In order to efficiently deliver the group key, the probability (P_c) that the key-sharing graph is connected must be as high as possible.

Using connectivity theory in a random-graph by Erdos and Renyi [17], we can obtain the necessary expected node degree *d* (i.e., the average number of edges connected to each node) for a network of size *N* when *N* is large in order to achieve a given global connectivity, P_c:

$$d = \frac{(N-1)}{N}\left[\ln(N) - \ln(-\ln(P_c))\right] \quad (12)$$

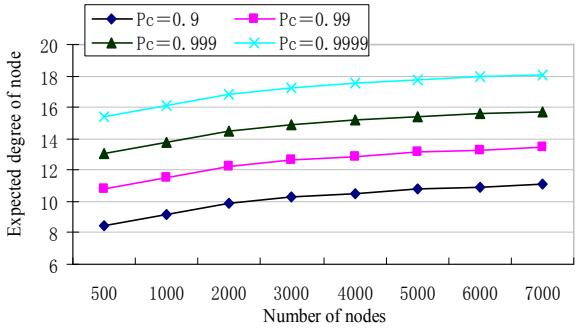

Figure 2. Expected degree of a node for varying number of nodes

Fig.2 illustrates the plot of the expected degree of a node, *d*, as a function of the network size, *N*, for various values of P_c. For example, we choose *N*=4000, to obtain P_c=0.999, the necessary expected node degree *d* is at least 16.

For a given density of network deployment, let *n* be the expected number of neighbors within the communication range of a node. Using the expected node degree calculated above, the required local connectivity, $P_{required}$, can be estimated as follows, $P_{required} = \frac{d}{n}$. After we have selected values for *q* and *m*, the actual local connectivity is determined by these values. We use P_{actual} to represent the actual local connectivity, which is the probability of any two neighboring nodes sharing at least one key. In our scheme,

$$P_{actual} = \frac{\frac{bk}{v} - 1}{b-1} \cdot k = \frac{k(bk-v)}{v(b-1)} = \frac{k}{q+1} \quad (13)$$

In order to achieve the desired global connectivity P_c, we should have $P_{actual} \geq P_{required}$, and make P_{actual} become as high as possible. According to equation (13), we observe that P_{actual} increases with increment of *k* for fixed *q*. When $k=q+1$, $P_{actual}=1$, namely, any pair of nodes can find at least a common key between them.

In Fig.3, we compare the P_{actual} of PDGRS and GKMPAN by varying *m*, the number of keys in a node. In PDGRS, *q*=113. And the key pool size of the two schemes is equal. We can observe that the P_{actual} of them with *m* increment, but PDGRS outperforms GKMPAN. That is, the P_c for PDGRS is much higher than that of GKMPAN with *m*.

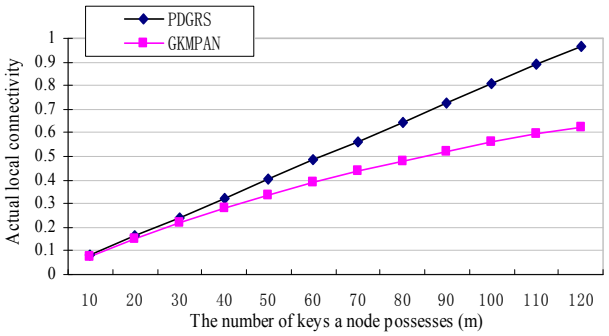

Figure 3. Comparison of the connectivity of the proposed scheme with GKMPAN scheme

V. CONCLUSIONS

Secure group re-keying has become an important component of many applications in wireless sensor networks. In this paper, we have presented PDGRS, a new t-packing design based group re-keying scheme for sensor networks, which focuses on key distribution and update for secure group communication. Apart from the previous approaches, we use Latin squares to construct orthogonal arrays in order to quickly obtain t-packing designs, which are adopted in key pre-distribution phase, and then the pre-deployed keys are used for group re-keying. The proposed scheme achieves cover-free family properties. Furthermore, the collusion-resilience as well as the key-sharing connectivity of networks improves with the increasing number of the keys in a node. Finally, updating pre-deployed keys can further enhance the security of the new scheme.

ACKNOWLEDGMENT

This work is supported Partially by Natural Science Foundation of China (NO.60502047), Key Science Foundation of Fujian High University in China (NO.JA07030), and Natural Science Foundation of Fujian Province (NO.2008J0014)

REFERENCES

[1] Akyidiz IF, Su W. Sankarasubramaniam Y. Cayirci E. Wireless sensor networks: A survey. Computer Networks, 2002, 38(4): 393-422.
[2] J. Kohl, and B. Neuman. The Kerberos Network Authentication Service, RFC 1510, September, 1993.
[3] Yu-Kwong Kwok. Key Management in Wireless Sensor Networks. Security in Distributed and Networking Systems. World Scientific Press. 2007. pp 75-98.
[4] G. Ateniese, M. Steiner, and G. Tsudik. a "New multiparty authentication services and key agreement protocols", IEEE Journal on Selected Areas in Communications, 18, no. 4, p. 628-640, April 2000.
[5] Li XU, Zhiwei Lin. Minimal Connected Dominating Set based Efficient Hybrid Key Management Strategy in Ad Hoc Network. The 2nd International Conference on Complex Systems and Applications. 2007.6 Press in DCDIS Series B, Vol.14 (s2), 1228-1231, Watam Press. Canada.
[6] Y. Kim, A. Perrig, and G. Tsudik. Communication efficient group key agreement, in proc. IFIP SEC'01, 2001.
[7] D. A. McGrew, and A. T. Sherman. Key establishment in large dynamic groups using one-way function trees, May 1998.
[8] L. Eschenauer and V. D. Gligor.: A key-management scheme for distributed sensor networks: Proceeding of the 9th ACM Conference on Computer and Communication security, (2002) 41-47
[9] H. Chan, A. Perrig, and D. Song. Random key pre-distribution schemes for sensor networks. IEEE Symposium on Security and Privacy, (2003) 197-213
[10] A. C.-F. Chan and E. S. R. Sr. Distributed symmetric key management for mobile ad hoc networks. In Infocom 2004.
[11] J. Wu and R. Wei. Comments on \distributed symmetric key management for mobile ad hoc networks" from infocom 2004. Cryptology ePrint Archive, Report 2005/008, 2005. http://eprint.iacr.org/.
[12] Zhu S, Setia S, Xu S, and Jajodia S. GKMPAN: An Efficient Group Rekeying Scheme for Secure Multicast in Ad-Hoc Networks[C]. In Proc. of International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004:42–51.
[13] D. Wagner, "Cryptanalysis of an Algebraic Privacy Homomorphism", ISC, pp. 234-239, 2003.
[14] Douglas R. Stinson, Ruizhong Wei: Generalized cover-free families. Discrete Mathematics 279(1-3): 2004: 463-477.
[15] Zhixu Yang. Construct of Orthogonal Arrays[M]. Jinang: Shangdong People Press, 1978.
[16] Lee B, Boyd C, Dawson E, Kim K, Yang J, and Yoo S. Secure Key Issuing in ID-Based Cryptography[C], CRPIT '04: Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation, Australian Computer Society, Inc., 2004: 69-74.
[17] Erdos, Renyi. On random graphs I. Publ. Math. Debrecen, 6:290–297, 1959.
[18] O. Goldreich, S. Goldwasser, and S. Micali. How to Construct Random Functions. Journal of the ACM, Vol. 33, No. 4, 1986, pp 210-217.
[19] Zhu S, Setia S, Xu S, and Jajodia S. GKMPAN: An Efficient Group Rekeying Scheme for Secure Multicast in Ad-Hoc Networks. Journal of computer Society, Volume 14, Number 4, 2006, pp.301-325.
[20] Wu, Zhongsheng; Chen, Zhide; Guo, Fuchun; Xu, Li. Identity Based Private Matching. The proceeding of Third International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing, 2007. July 2007. Pp: 85-90.
[21] San Ling, Huaxiong Wang and Chaoping Xing. Cover-Free Families and Their Application. Security in Distributed and Networking Systems. World Scientific Press. 2007. pp 75-98.
[22] W. Du, J. Deng, Y. S. Han and P. K. Varshney, A pairwise key pre-distribution scheme for wireless sensor networks, Proc. of the 10th ACM conf. on Computer and communications Security, (2003), 42-51.

Li Xu is a professor in the Department of Math and Computer Science at the Fujian Normal University. He received the B.S and M.S degrees form the Fujian Normal University in 1992 and 2001. He received the Ph.D. degree from the Nanjing University of Posts and Telecommunications in 2004. Now he is the assistant dean of School of Maths and Computer Science and Co-Director of Key Lab of Network Security and cryptography

He specializes in network protocol design, especially for wireless network. His current research focuses on secure issues in wireless ad hoc networks.

Dr. Xu is the senior member of CCF and CIE in China. He has published over 60 papers. His research group at FJNU has developed Network and Information Security, Complex System and Network, Intelligent Information Processing in Communication Networks, P2P, Grid, and Intelligent Information Processing in Communication Network. Contact him at Email: xuli@fjnu.edu

Jiangwei Chen is a lecture in the Department of Math and Computer Science at the Fujian Normal University. He received the B.S and M.S degrees form the Fujian Normal University in 2002 and 2007.

His research interests include mobile ad hoc networks and sensor network.

Xiaoding Wang is a assistant research in the key lab of security and cryptology of Fujian Normal University. He received the Master degree of Computer Science from Wollongong University of Australia in 2008.

His researches focus on cryptology and network security