A Hexagon-based Key Pre-distribution Scheme for Large Scale Static Wireless Sensor Networks

Xuanxia Yao
School of Information Engineering, University of Science and Technology Beijing, Beijing, China
Email: yaoxuanxia@163.com

Xuefeng Zheng and Tao Wu
School of Information Engineering, University of Science and Technology Beijing, Beijing, China
Email:zxfxue@263.net, wswuxihua@163.com

***Abstract*—In order to improve the secure connectivity & expansibility of static wireless sensor networks, decrease the memory costs of sensor nodes, an efficient hexagon-based key pre-distribution scheme is put forward by employing the ideas of the grouping key management and secret binding. In this scheme, the process of establishing pair-wise keys for neighboring nodes in the network is limited in the beginning of the network deployment, and when adding new sensor nodes into the network, the process of establishing pair-wise keys for the new nodes and their neighbors needs to be verified by the base-station. For the neighboring nodes in the same group, the polynomial-based key pre-distribution scheme is used to generate pair-wise keys for them. And for the neighboring nodes in different groups, the binding secrets generated by a HMAC are used to establish the pair-wise key. In addition, by analyzing the relations among the radius of the cell, the probability of the secure connection, memory costs and other parameters, the most appropriate value of the cell's radius is found, which can optimize the hexagon-based key pre-distribution schemes in the aspects of secure connectivity and memory costs.**

***Index Terms*—wireless sensor networks, key pre-distribution, hexagon-based model, secret binding, HMAC**

I. INTRODUCTION

With the wide application in many fields of human life ranging from military applications to civilian applications, the security of wireless sensor networks has received a lot of attention. Providing security services to wireless sensor networks can assure the security of them. Key management, as a fundamental security service, is very important for various security services, such as encryption and authentication. However, sensor nodes typically operate in unattended conditions and have limited computational capabilities, memory and battery-power capacity, which makes the materialization of the efficient key management schemes in wireless sensor networks become very difficult. It is obvious that asymmetric cryptographic algorithms are not suitable for wireless sensor networks. Many studies in recent years have indicated that the key pre-distribution scheme is a feasible key management scheme for wireless sensor networks because of its low requirements for resources, in which keys materials are pre-distributed among all nodes prior to deployment. According to reference [1], a key pre-distribution scheme is a method to distribute off-line private key materials among a set of users, such that each group with a given size can compute a common key for secure communication. Actually, in most of the applications of wireless sensor networks, long distance peer-to-peer communication is rare and unnecessary. And the primary goal of secure communication is to provide authentication and/or encryption between neighboring nodes that can directly communicate with each other. So the size of each group in our scheme is two and the task of our scheme is to establish pair-wise keys for two neighboring nodes.

As we know, the high probability of the secure connection, the good expansibility, the low costs and the strong resistibility against being captured are the characters that an efficient key pre-distribution scheme for wireless sensor network should have. In this paper, we try to partition the target fields according to the hexagon-based model, to bind the group secrets with the function of HMAC and to establish the pair-wise key for the neighboring nodes in the same group by the polynomial-based key pre-distribution scheme, which can make our proposed scheme realize the above four characters.

The rest of the paper is organized as following: section two describes the related works; section three gives the overview of the network model and the related denotations and conceptions; section four depicts the proposed scheme in detail; section five discusses the performance analysis; section 6 concludes the paper.

II. RELATED WORKS

A. The State of The Art

Presently, there are many key pre-distribution schemes for wireless sensor networks have been put forward. And

these key pre-distribution schemes can be classified into four kinds, that is probabilistic schemes, deterministic schemes, hybrid schemes and location aware or group-based schemes.

In probabilistic key pre-distribution schemes, key chains are randomly selected from a pool and distributed to sensor networks, such as the random key chain pre-distribution [2], the random pair-wise key scheme [3] and the adaptive random key distribution scheme [4]. In a general way, probabilistic approaches usually have good expansibility, but they are with low secure connectivity. Deterministic key pre-distribution approaches are used to provide better connectivity, such as the matrix-based scheme [1], the polynomial-based key generation scheme [5] and the SBIBD-based scheme [6] as well. The storage costs of deterministic schemes are all relatively high and hard to support large size wireless sensor networks. Furthermore, most of these schemes have very poor expansibility, such as [1] and [6]. Hybrid key pre-distribution approaches try to use probabilistic approaches on deterministic solutions to improve the expansibility and decrease the memory costs, such as the polynomial pool based scheme [7], the threshold based scheme [8] and the adaptive random key distribution scheme. But these hybrid schemes can not settle the inherent problems in probabilistic approaches and deterministic approaches yet. Location-aware or group-based key pre-distribution schemes are developed in recent year, such as the group-based key pre-distribution schemes [9] [10] and the key pre-distribution schemes by using deployment knowledge [11] [12]. They can provide better connectivity, expansibility and reliability by using the prior and/or post deployment knowledge of the sensor nodes and/or managing the nodes in groups. Although these schemes have many good characters, the existing location-aware or group-based key pre-distribution schemes still need relative more memory costs, which is the issue that we will discuss in this paper.

Our scheme combines the group-based key pre-distribution approaches with the polynomial-based scheme and employs HMAC function to bind the node with the secrets of its adjacent groups, which can help large scale wireless sensor networks to realize the high secure connectivity, the good expansibility, the low costs and the strong resistibility against being captured.

B. Group-Based Key Pre-Distribution Scheme

The theory foundation of group-based key pre-distribution schemes is that the nodes in wireless sensor networks only communicate with their neighbors due to the limited batter-power capacity and the small communication range of nodes [9]. On the basis of this theory, the large area that the wireless sensor network covered can be divided into small sub-areas or cells, accordingly, the nodes in the network should be partitioned into groups as many as cells. And one group is corresponding to one cell. All the nodes in one group are distributed the key materials in the same way and are expected to be deployed in the corresponding cell of the group. According to the analysis of Bo Yu etc [13], most of the nodes in one group only communicate with its neighbors in the same group, especially when the cell's width or radius is relatively greater than the signal range of a node. So the nodes in the same group have the high probability of neighboring, and the two neighboring nodes have the high probability of establishing pair-wise key too. These characteristics can improve the secure connectivity and enhance the expansibility.

Nowadays, the studies on group-based key pre-distribution schemes are focusing on how to partition the target field and establish the pair-wise key for the neighboring nodes in different groups.

As for the issue of partitioning the target field, the grid-based model [14] and the hexagon-based model [15] are the two mainly models. The grid-based model is a simple model, which can correspond to the common rectangular coordinate system easily but can not reflect the character of wireless broadcasts well. The hexagon-based model can simulate the signal propagation and reflect the wireless broadcasts well, furthermore, the relations among cells are symmetric and taking on hierarchy frame, which can make us denote the locations relations of cells conveniently. For example, we can say that one cell is lying at the n^{th} layer of the other cell. In Fig. 1, for cell C_3, cell C_0, C_5 and C_{18} are locating at its first, second and third layer respectively. On the contrary, C_3 locates at the first layer of C_0, the second layer of C_5 and the third layer of C_{18}. In this scheme, the hexagon-based model is used to partition the target field.

As for establishing the pair-wise key for the neighboring nodes in different groups, almost all of the existing group-based key pre-distribution schemes employed the polynomial-based key pre-distribution scheme. In the polynomial-based key pre-distribution scheme, a bivariate t-degree symmetric polynomial $f(x,y)$ over a finite field F_q is used to establish pair-wise key for

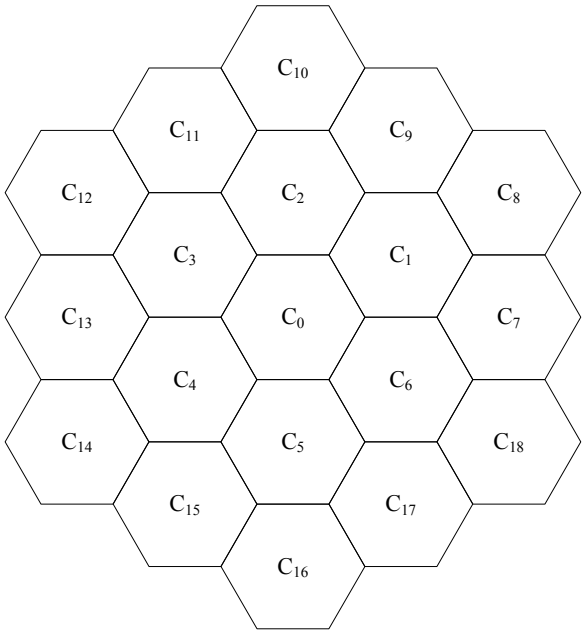

Figure 1. Hexagon-based model

two nodes, where q is a prime number that is large enough to accommodate a cryptographic key. If only there are no more than t compromised sensor nodes, the pair-wise keys between any two non-compromised sensor nodes are security or not known to the compromised sensor nodes. In addition, the polynomial-based key pre-distribution scheme has good secure connectivity. It is only fit for small scale wireless sensor networks due to its high memory costs and the character of t-collision resistant. In our scheme, the polynomial-based key pre-distribution scheme is only used to establish pair-wise key for the nodes in the same group. And for the nodes in different groups, the HMAC function is employed to bind the secrets so as to establish pair-wise keys for them and save the memory.

III. The Related Explanations

A. Network Model

In this scheme, the network is assumed to be a static large scale wireless sensor network and be deployed on the basis of the hexagon model. The target field is divided into some small, equal-size hexagons according to the requirements of the application, and these hexagons are the so-called cells. Similarly, all nodes in the network are partitioned into groups as many as hexagons. And one group has one corresponding hexagon or cell. Nodes in one group are expected to be deployed in its corresponding cell. If the side length or radius of each cell is R and the maximum deployment error is e, the nodes in one group will locate in the circle area whose center is the center of the group's corresponding cell, and its radius is $(R+e)$.

At the same time, we also assume that the sensor nodes in the network are distributed uniformly over the target field and the links are bidirectional. All the nodes are low power and can only communicate with their neighboring nodes. The communication range, computing ability, memory size and the status in the network of all sensor nodes are identical. And all the neighboring relations are mutual. That is, if X is a neighbor of Y, Y is also a neighbor of X.

In addition, the network is also assumed to include a globally trusted base station, which has enough resources and knows all the key materials of each group and each node. Furthermore, the base station is the ultimate destination for the data sensed from the network and also responsible for initiating notice to the sensor nodes in the network, updating the network and so on.

B. Attack Model

There are many attacks on wireless sensor networks. But if the security services have been provided with a wireless sensor network, the most important thing for the attacker who wants to control the network should be destroying the security services implanted on it. So launching attacks on the key management scheme of the network is the main task of the attackers. Although there are different attack methods to different key management schemes, almost all of the attack methods need to capture some nodes to obtain the key materials in them and get the way of establishing the pair-wise key so as to deduce the shared-key between any two nodes. Therefore, the best way to defend this attack is to let the nodes destroy the key materials of themselves when being captured, which can make the nodes very expensive and is not practical. What we can do is to leak the key materials as few as possible.

In this paper, we consider the fact that the attackers have not enough time to attack at all in the beginning of the network deployment. Even they have captured some nodes, analyzing the data in the captured nodes also expends a little time. So considering the security of the network, we limit the process of the initial pair-wise key establishment in the beginning of the network deployment, and all the related key materials should be deleted after the pair-wise key being established.

C. The Related Notations and Conceptions

In order to describe the scheme conveniently, some notations and conceptions used in this scheme are illuminated as following:

G_i: Group i.

C_i: Cell i, C_i is corresponding to G_i.

$<GID_A, NID_A>$: The global identifier of node A, where GID_A is the identifier of the group that node A belongs to, and NID_A is the local identifier of node A in the group.

$K_{A,B}$: The pair-wise key between node A and node B.

$K_{A,B}'$: The temporary pair-wise key between node A and node B.

$E_{Ks}(M)$: The message M is encrypted by key K_s.

K_A: The unique master key shared by node A and the base station, which is used to assure the security of the communication between them.

KG_i: The group key of G_i, which is shared by all the nodes in G_i.

K: The network key, which is shared by all the nods in the network and should be refreshed periodically.

$H(x,y)$: A HMAC function, which represents performing hash operation on y with the key x.

HK_A: The hiding master key of node A, which is defined as

$$HK_A=H(K_A, K_A). \quad (1)$$

MK_A: The mask value of node A's hiding mater key, which is used to assure that the hiding master key is only known to the rightful nodes in the beginning of the network deployment and defined as

$$MK_A =(HK_A \oplus K). \quad (2)$$

$F_i(x,y)$: The bivariate t-degree symmetric polynomial over finite field F_q of G_i.

$KG_{A,i}$: The binding secret of node A and group G_i, which is defined as

$$KG_{A,i}=H(KG_i, HK_A). \quad (3)$$

Conception 1: Home cell. One node's home cell is the cell corresponding to the group that it belongs to.

Conception 2: Neighboring cells. If one cell locates at the first layer of the other cell, we call the two cells are neighboring cells.

Conception 3: Neighboring groups. If one group's corresponding cell and the other group's corresponding cell are neighboring cells, we call the two groups are neighboring groups.

Ⅳ. The Proposed Scheme

The proposed scheme includes three phases. They are initialization phase, shared-key establishing phase and the updating phase.

A. Initialization

Initialization operations are accomplished by setup server before nodes deployment. There are five things to be done in the initialization phase.

The first thing is to decide the radius R of the cell and the secret binding layer n according to the security requirements of the application, the maximum deployment error e, the size of the target fields, the communication range and the memory size of the node.

The second thing is to divide the target field into z hexagon with the radius R and partition all the nodes into z groups. One group is corresponding to one cell.

The third thing is to generate a unique bivariate t-degree symmetric polynomial over a finite field F_q for each group.

The fourth thing is to assign key materials for each node. In this scheme, the key materials include network key, master key, the share of the polynomial, the binding secrets and so on. The former two can be generated directly by the setup server, and the latter two need to be evaluated by the setup server. For node A in group G_i, the share of its polynomial is computed as: $F_i(GID_A \oplus NID_A, y)$. And the binding secret is computed according to (3).On the assumption that the secret needs to be bound to layer n, all the groups whose corresponding cells locate in the area from the first layer of one node's home cell to the n^{th} layer of it should be bound with the node. According to the hexagon model, such groups come to $(3n^2+3n)$. The symmetric relation of their locations makes the binding secret have the character of symmetry. For instance, if the nodes in G_i have the binding secret with group G_j, the nodes in G_j have the binding secret with group G_i too.

The fifth thing is to set a timer for each node so as to limit the process of the initial shared-key establishment in the beginning of the network deployment.

B. Pair-Wise Key Establishment

In this scheme, the process of establishing pair-wise key for neighboring nodes is the process of discovering the neighbors.

1) Each node starts its timer and broadcasts its neighbor discovery requirement after deployment. The neighbor discovery requirement consists of the node's global identifier and the mask value of the hiding mast key. For example, the neighbor discovery requirement of node A is "$GID_A \| NID_A \| (HK_A \oplus K)$".

At the same time, every node also receives the neighbor discovery requirements from its neighbors and establishes its neighbor list according to the received neighbor discovery requirements.

2) The main task of this step is to establish the pair-wise key for each node in its neighbor list. For a node A in group G_i, node B is assumed to be one node of its neighbor list, the process of establishing a pair-wise key for node A and B is described as following:

If $GID_A=GID_B$, the polynomial-based key pre-distribution scheme is used to compute the pair-wise key for them. According to the polynomial-based key pre-distribution scheme, $K_{A,B}$ is computed as

$$K_{A,B}=F_i((GID_A \oplus NID_A),(GID_B \oplus NID_B))$$

$$= F_i((GID_B \oplus NID_B),(GID_A \oplus NID_A))= K_{B,A}. \quad (4)$$

If $GID_A \neq GID_B$, it is assumed node B is belong to group G_j, there are two cases. The first case is that G_j is beyond the secret binding range of node A and the two nodes have no binding secret with each other's group, it is unable to establish pair-wise key for them directly and they just delete the other side from their own neighbor list respectively. In fact, according to the first operation in the initialization phase, this case should be rare. The second case is that G_j is within the secret binding range of node A, which is a very frequent case. According to the fourth operations in the initialization phase, node A has $KG_{A,j}$ and node B has $KG_{B,i}$. Furthermore, since the neighbor discovery requirement of one node includes the mask value of its neighbor's hiding mast key, node A can compute $KG_{B,I}$ according to equation(5).

$$KG_{B,i}= H(KG_i,(HK_B \oplus K) \oplus K)= H(KG_i,HK_B) \quad (5)$$

And node B can compute $KG_{A,j}$ according to equation (6).

$$KG_{A,j}=H(KG_j,(HK_A \oplus K) \oplus K)=H(KG_j,HK_A) \quad (6)$$

Well then, the temporary pair-wise key $K_{A,B}'$ between node A and node B is computed as

$$K_{A,B}'= KG_{B,i} \oplus KG_{A,j}= KG_{A,j} \oplus KG_{B,i}=K_{B,A}'. \quad (7)$$

In addition, for the sake of avoiding the threats caused by the leak of hiding master key, a new pair-wise key should be negotiated immediately by the two neighboring nodes in different group after the temporary pair-wise key is established. A simple way is:

Let node A generate a random number r_A as the new pair-wise key $K_{A,B}$, encrypt r_A with $K_{A,B}'$ and send it to node B. After receiving the message, node B decrypts the message to get r_A as their common key $K_{A,B}$ too.

3) All the binding secrets, the hiding master keys of all neighbors and the middle key materials are deleted.

C. Network Updating

Network updating includes appearing failure nodes in the network and adding new sensor nodes into the network. The first case is simple. Once a node detects or receives the notice that one of its neighbors is failure or

dead, it just deletes the node from its neighbor list and broadcasts the failed node to its other neighbors. The second case that adding new sensors into network is more complicated. The instance of adding a new sensor node C into group G_i will be illustrated as following.

Firstly, node C should be initialized before being added into the network. At the same time, the base station should inform the nodes in the adjacent groups of group G_i that node C will be deployed in group G_i or cell C_i. After being deployed in the network, Node C will start a timer and generate a random r_C and perform the XOR operation on r_C and the hiding master key HK_C. And then, the neighbor discovery requirement is constructed and broadcasted. The neighbor discovery requirement is described as following: "$GID_C||NID_C||$（$r_C \oplus HK_C$）".

For the node D that received the neighbor discovery requirement, if node D is not a new sensor node and did not receive the notice about adding node C from the base station in advance, node C is considered an attack node and node D will do nothing about the requirement. Otherwise, node D will have to process this requirement. There are two cases.

The first case is $GID_D=GID_C$. Node D computes the shared key $K_{D,C}$ according to the global identifier of node C and its share of the polynomial, as in

$$K_{D,C}=F_i((GID_D \oplus NID_D),(GID_C \oplus NID_C)). \quad (8)$$

Afterwards, node D sends the message "$GID_D||NID_D||\ E_{K_{D,C}}(GID_D \oplus NID_D)$" to node C. Node C computes $K_{C,D}$ according to the global identifier of node D and its share of the polynomial after receiving the message, and decrypts "$E_{K_{D,C}}(GID_D \oplus NID_D)$"to verify the shared-key $K_{C,D}$.

$$K_{C,D}=F_i((GID_C \oplus NID_C),(GID_D \oplus NID_D)) \quad (9)$$

The second case is $GID_D \neq GID_C$. In this case, node D generates a random r_D and sends the message "$E_{K_D}(r_D \| GID_C \| NID_C \| (r_C \oplus HK_C))$" to the base station. The base station decrypts the message "$E_{K_D}(r_D \| GID_C \| NID_C \| (r_C \oplus HK_C))$" after receiving it. And then the base station recovers r_C from "($r_C \oplus HK_C$)" by computing HK_C with K_C and performing XOR on "($r_C \oplus HK_C$)". Subsequently, $KG_{D,i}$ can be computed as

$$KG_{D,i}=H(KG_i,(r_D \oplus HK_D)). \quad (10)$$

After getting $KG_{D,i}$, the message "$E_{K_D}(r_C \| KG_{D,i})$" is constructed and sent to node D. Node D decrypts the message and gets r_C and $KG_{D,i}$. Here, HK_C can be obtained by performing XOR operation on r_C and ($r_C \oplus HK_C$). So $KG_{C,j}$ can be computed as

$$KG_{C,j}= H(KG_j,\ HK_C). \quad (11)$$

Accordingly, the shared-key $K_{C,D}$ can be evaluated as

$$K_{C,D}=KG_{D,i} \oplus KG_{C,j}. \quad (12)$$

And then node D sends message "($r_D \oplus HK_D$)" to node C. After receiving the message, node C can compute $KG_{D,i}$ as

$$KG_{D,i}= H(KG_i,(r_D \oplus HK_D)). \quad (13)$$

If node C have the binding secret with group G_j, it can evaluate $K_{C,D}$ in the form of equation (14).

$$K_{C,D}=KG_{D,i} \oplus KG_{C,j} \quad (14)$$

Otherwise, node C makes $KG_{D,i}$ as $K_{C,D}$ and sends the message "$E_{K_{C,D}}(r_C+1)$" to node D. Node D also makes $KG_{D,i}$ as $K_{C,D}$ by decrypting the message with $KG_{D,i}$ and comparing the decrypted message with "(r_C+1)".

It can be seen that the new rightful sensor nodes can always establish the pair-wise keys with its neighbors.

Ⅴ. Performance Analysis

A. Probability of Establishing Pair-Wise Key.

In order to compute the probability of establishing a common key between any two neighbor sensor nodes, it is necessary to get the number of a node's neighbors and the number of the neighbors that can establish the common key with the node. Considering any two groups G_i and G_j, it is assumed that the coordinates of their home cell C_i and C_j are (x_i,y_i) and (x_j,y_j) respectively. The probability that the node in group G_i is border upon the node in group G_j can be calculated by equation (15).

$$p(G_i,G_j)=\begin{cases}0, d \geq (2R+2e+dr) \\ \dfrac{1}{\pi(R+e)^2} \displaystyle\iint_{\substack{d1\leq(R+e+dr)\text{and} \\ d2\leq(R+e)}} f(x,y)dxdy, dr<d<(2R+2e+dr)\end{cases} \quad (15)$$

Here, $f(x,y)$ is defined as equation (16). And d, $d1$, $d2$ are defined as equation (17), (18), (19) respectively.

$$f(x,y)=\begin{cases}\dfrac{dr^2}{(R+e)^2}, d1 \leq (R+e-dr) \\ 1, d1 \leq (dr-(R+e)) \\ \dfrac{1}{\pi(R+e)^2} \displaystyle\iint_{\substack{\sqrt{(x'-x)^2+(y'-y)^2}\leq dr;\text{and} \\ \sqrt{(x'-x_j)^2+(y'-y_j)^2}\leq(R+e)^2}} dx'dy', |(R+e-dr)|<d1<(R+e+dr) \\ 0, d1 \geq (R+e+dr)\end{cases} \quad (16)$$

$$d=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \quad (17)$$

$$d1=\sqrt{(x-x_j)^2+(y-y_j)^2} \quad (18)$$

$$d2=\sqrt{(x-x_i)^2+(y-y_i)^2} \quad (19)$$

It can be seen from the formula of $p(G_i,G_j)$ that after the nodes' signal range dr, maximum deployment error e and the radius of the cell R are determined, the probability of two sensors neighboring is depending on d,

which is the distance between the two home cells' centers of the two nodes. It is easy to expressed d as the expression of layer n and R according to the hexagon model. So the probability of two nodes neighboring in any two groups can be calculated in the way of gauss integral.

It is assumed that there are m nodes on average in each sensor's range. The density of the network w can be estimated by

$$w = \frac{(m+1)}{\pi \cdot dr^2}. \tag{20}$$

According to the hexagon model, the area of each cell is $\frac{3\sqrt{3}}{2}R^2$, so the number of nodes in each cell on average can be evaluated by

$$N_C = \frac{3\sqrt{3}}{2 \cdot \pi \cdot dr^2} \cdot (m+1) \cdot R^2. \tag{21}$$

For all nodes in group G_j, the average number of nodes that a node u in group G_i can directly communicate with can be estimated by N_C*$p(G_i,G_j)$, therefore, the average number of nodes that a node u in group G_i can directly communicate with can be estimated by

$$n_u = N_C \cdot \sum_{\forall j} p(G_i, G_j). \tag{22}$$

In order to compute the number of the neighboring nodes that can establish the common key with node u, the neighboring relations is further divided into neighboring in one group and neighboring between two different groups. For the two neighboring nodes in one group, they can establish a pair-wise key according to the shared bivariate t-degree polynomial. For the two neighboring nodes in two different groups respectively, establishing a pair-wise key depends on whether they have the binding secret of the group that the other node belongs to, which is determined by the number of the layers that be bound secrets. We can see from the equation of $p(G_i,G_j)$ that the probability of two nodes neighboring is 0 when the distance d between the center of their home cell is more than $(2R+2e+dr)$. That is to say that if two nodes are neighboring, the distance d between the center of their home cell should be less than $(2R+2e+dr)$. And d is a function with n and R as its parameters. So we can deduce that if a node is a neighbor of node u, whose home cell locates at most at the layer of n_m^{th} out of the home cell of node u. And n_m is a function with dr, e and R as its parameters, where dr is fixed when the sensors are selected, and we assume that e is equal to dr, then n_m is only decided by R. The relation between R and n_m is shown as Fig. 2.

Let S_i denotes the set of the home cells that having binding secrets with the nodes in group G_i. According to the hexagon model, if the number of the binding layer is n, there are $(3n^2+3n)$ cells in set S_i. So the number of the sensors that can establish pair-wise key with node u can be evaluated by equation (23).

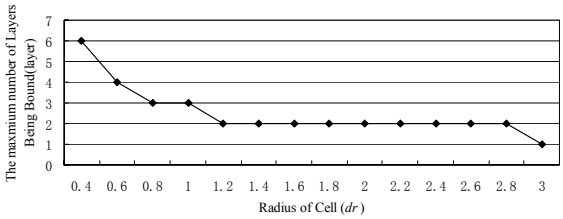

Figure 2. The relation between R and n_m

$$n_u^s = N_C \cdot \left(\sum_{G_j \in S_i} p(G_i, G_j) + p(G_i, G_i) \right) \tag{23}$$

The probability of establishing a common key between any two neighboring sensor nodes can be estimated by equation (24).

$$p = p(u) = \frac{n_u^s}{n_u} \tag{24}$$

It is obvious that p is determined by n and R, and the relations among them is shown in Fig. 3.

It also can be seen from Fig. 3 that if n is fixed, p tends towards 1 with the increasing of R. If R is given a certain value, p is increasing with n rapidly. When n is equal to n_m, p is up to 1. In practice, the best value of n and R should be found according to the memory and dr of the node so as to make p be close or equal to 1.

B. Memory Costs

For the convenience of analyzing the memory costs, we divide the memory costs into the variable memory costs and the constant memory costs. The variable memory costs changes with other parameters, such as p. In this scheme, they are the memory for storing the binding secrets and the memory for storing the share of the bivariate t-degree polynomial. The constant memory costs doesn't change with other parameters after the security parameters are fixed, such as the memory for storing the network key, master key, group key and so on. So it only needs to analyze the variable memory costs when analyzing the relations among the memory costs and other parameters, but both the variable memory costs and the constant memory costs should be considered when making comparison among different schemes. In this paper, the relations among the memory costs and other parameters are analyzed in advance so as to optimize the scheme, and then the comparison between

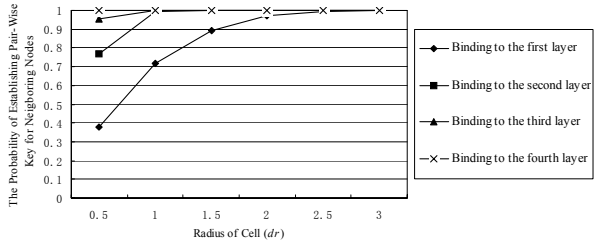

Figure 3. The relations among p, n and R

our scheme and the scheme based on polynomial fully is made.

For the nodes in the same group, the polynomial-based key pre-distribution is used to establish pair-wise key, every node needs to store the share of the bivariate t-degree polynomial. If the length of the key is 128 bits, the memory costs for storing the share of the bivariate t-degree polynomial can be computed by

$$mv1=(t+1)*\log 2^{128}=128*(t+1). \quad (25)$$

It is assumed that the value of the HAMC function is 128 bits, the memory to store a binding secret is 128 bits too. If the secret is bound to layer n^{th}, there are $(3n^2+3n)$ binding secrets to store and the memory to store the binding secrets is evaluated by

$$mv2=128*(3n^2+3n). \quad (26)$$

The total variable memory costs are estimated by

$$mv=mv1+mv2=128*(3n^2+3n)+ 128*(t+1). \quad (27)$$

According to the foregoing analysis, n is the function of R. The degree of the polynomial t is determined by the number of the nodes in one cell. In order to enhance the security of the polynomial, t is set by $(1.5N_C)$. The relations among the variable memory costs, the average number m of the neighbors and the radius R of the cell are shown in Fig. 4.

Because the condition that the radius R of the cell is less than the node's signal range dr will make the group-based key pre-distribution scheme lose its grouping significance, in this paper, we only consider the condition that R is great than dr. Obviously, no matter what the density of the network is, the variable memory cost is the least when R is about equal to $1.2dr$. In addition, it also can be seen from Fig. 2 that when R is equal to $1.2dr$, the probability of establishing pair-wise key for the neighboring nodes can be up to 1 if only they are bound to the second layer. So it can be educed that $R\approx1.2dr$ is the best point for the security connectivity and the memory costs.

Similar to the general security scheme, the one-way hash function is also used in this scheme, but the hash function used in this scheme works according to different keys for different purposes. So there is no need to consider the costs of the HMAC function when comparing with other schemes. And the constant memory costs in this scheme are basically equal to the constant memory costs in the polynomial-based scheme. The difference between their memory costs is only the variable memory costs. When the average number m of neighboring nodes is 15, the memory costs in the two schemes are shown in Fig. 5.

It can be seen that the memory costs in our scheme is far less than the memory costs in the scheme based on polynomial fully.

C. Security Analysis

In this scheme, there are two cases of establishing pair-wise keys. The first case is to establish the pair-wise key in the beginning of the network deployment and the second case is to establish pair-wise key when adding the new sensor node into the network. In the first case, the pair-wise keys are established by exchanging the neighbor discovery requirements among the neighboring nodes. And in the second case, the process of establishing pair-wise key needs the verification of the base-station in order to realize expandability and improve the security. At the same time, for avoiding speculating and forging, the binding secret makes the hiding master key as the parameter. And the hiding master key, binding secret and other secret materials should be deleted as soon as the initial pair-wise key is established. Furthermore, the initial pair-wise keys between two nodes in different group should be changed immediately so as to enhance the security of the key and decrease the probability of being attacked. In addition, the degree of the polynomial for each group is set *1.5* times of the average number of neighboring nodes so as to make the attacker can not recover the polynomial even if they have captured all the nodes in one group.

In conclusion, one captured node can only effect the communication between it and its neighboring nodes, and can not effect the communications among other nodes. This can limit the attack in the communication range of the captured node and make the network more secure.

D. Communication and Computation Costs

In this scheme, the communication costs include three aspects. The first is for interchanging the neighbor discovery requirements between neighboring nodes in the phase of initial pair-wise key establishment. The second is for updating and validating the initial pair-wise key between neighboring nodes in different group. The third is to verify the new node with the base station in the process of running. So the communication costs in our scheme are more than that in the scheme based polynomial fully.

The computing costs in this scheme are relatively little. All operations are simple, such as evaluating the value of the polynomial, performing hash or XOR operations.

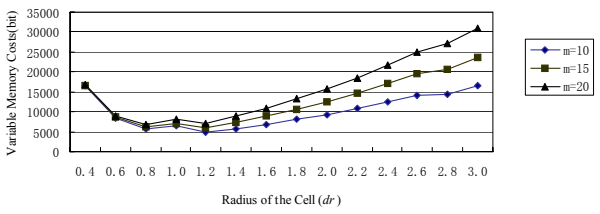

Figure 4. The relations among mv, R and m

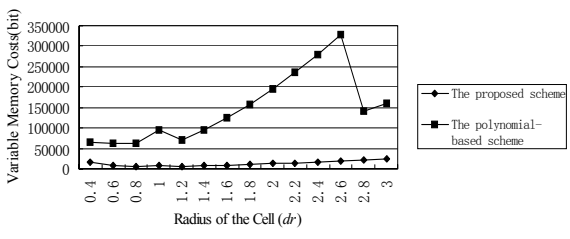

Figure 5. The memory costs of the two schemes when m=15

VI. CONCLUSIONS

The grouping key distribution scheme for static wireless sensor networks is put forward in this paper by employing the grouping key management technology, the method of secret binding and the hexagon-based model, which has the characters of better secure connectivity, expandability, the ability of defending attacks. The best value of R is fund by analyzing the relations among R and each parameter, which can optimize the hexagon-based key pre-distribution schemes in the aspects of secure connectivity and memory costs. In addition, as for the security of wireless sensor networks, our works also include the secure routing, the trust model, broadcast authentication and so on.

REFERENCES

[1] R.Blom. "An Optimal Class of Symmetric Key Generation Systems." *In Advances in cryptography: Proceedings EUROCRYPT*, 1985.

[2] L.Eschenauer and V.D. Gligor."A Key Management Scheme for Distributed Sensor Networks" , *in Proceedings of 9th ACM Conference on Computer and Communications Security*, Washington, DC, USA, ACM Press, November 2002, p41-47.

[3] H.Chan, A.Perrig, and D.Song. "Random Key Pre-Distribution Schemes for Sensor Networks", *In Proc. IEEE Symposium on Research in Security and Privacy (SP 2003)*, 2003, p197-213.

[4] S.Y. Wu and S.P. Shieh. "Adaptive Random Key Distribution Schemes for Wireless Sensor Networks",*in Proceeding of 2003 Int'l Workshop on Advanced Developments in Software and System Security*, Dec.2003, p61-65.

[5] C.Blundo, A.De Santis, A.Herzberg, S.Kutten, U.Vaccaro, and M.Yung. "Perfectly Secure Key Distribution for Dynamic Conferences", *Advances in Cryptology - CRYPTO '92*, LNCS 740, 1993, p471-486.

[6] Seyit A. Çamtepe, Bülent Yener. "Combinatoral Design of Key Distribution Mechanisms for Wireless Sensor Networks", *EUROSICS(2004)*.

[7] D.Liu and P.Ning. "Establishing Pair-Wise Keys in Distributed Sensor Networks",*in Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*, October 2003, p52-61.

[8] A. Price, K. Kosaka, and S. Chatterjee. "A Secure Key Management Scheme for Sensor Networks." *In Proceedings of the 10th Americas Conference on Information Systems*, New York, August 2004.

[9] Donggang Liu, Peng Ning, Wenlisng Du. "Group-Based Key Pre-Distribution in Wireless Sensor Networks", *in Proceedings of the 4th ACM workshop on wireless security*, Cologne Germany: ACM Press, 2005, p11-20.

[10] Po-Jen Chuang, Tun-Hao Chao, Bo-Yi Li. "A Scalable Grouping Random Key Pre-distribution Scheme for Large Scale Distributed Sensor Networks", *in Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05)*,volume 2, July 2005, p535-540.

[11] W. Du, J. Deng, Y. S. Han, S. Chen, and P. Varshney. "A Key Management Scheme for Sensor Networks using Deployment Knowledge." *In Proceedings of IEEE INFOCOM'04*, March 2004.

[12] Donggang Liu, Peng Ning. "Improving Key Predistribution with Deployment Knowledge in Static Sensor Networks", *ACM Transactions on Sensor Networks*,Vol. 1, No. 2, November 2005, p204-239.

[13] Bo Yu, Xiaomei Cao, Peng Han, Dilin Mao, Chuanshan Gao. "Flexible Deployment Models for Location-Aware Key Management in Wireless Sensor Networks", *APWeb 2006*, LNCS 3841, p343-354.

[14] Donggang Liu, Peng Ning. "Location-Based Pair-wise Key Establishments for Static Sensor Networks", *in Proc. 2003 ACM Workshop on Security in Ad Hoc and Sensor Networks*.2003, p52-61.

[15] Guorui Li, JingSha He, and Ying Fang Fu. "Key Pre-Distribution in Sensor Networks", *UIC2006*, LNCS 4159,2006, pp 845-853.

First Xuanxia. Yao was born in Luoyang on 17th, March,1971. She got her bachelor degree in computer science in 1994 from Jiangsu University. In 2002, she earned the master degree in computer science from the University of Science & Technology Beijing. And now she is a PH.D candidate in the University of Science & Technology Beijing, her major is network and information security.

She worked as a SOFT ENGINEER in from 1994 to 1999 and works as a LECTURE from 2002 to now. Her Current research interests is network security and wireless sensor network.

Ms. Yao is a memberships in China Computer Federation.

Second Xuefeng. Zheng was born in Fuzhou, Jan, 1951. Now he is a professor in department of computer science, the University of Science & Technology Beijing. His major is network security. Mr zheng is a senior member of China Computer Federation.

Third Tao. Wu was born in Shanxi Province in 1974. She got her PH.D in Jan, 2008. Her major is information security. Now she works as a Lecture in China Women's University.