A Location-aware Key Predistribution Scheme for Distributed Wireless Sensor Networks

Ngo Trong Canh[1], Tran Van Phuong[1], Young-Koo Lee[1*], Sungyoung Lee[1], and Heejo Lee[2]
[1]Dept of Computer Engineering, Kyung Hee University, Korea.
[2]Dept of Computer Science and Engineering, Korea University, Korea.
{ntcanh, tvphuong}@oslab.khu.ac.kr, yklee@khu.ac.kr, sylee@oslab.khu.ac.kr, heejo@korea.ac.kr

***Abstract*-Key establishment plays a central role in authentication and encryption in wireless sensor networks, especially when they are mainly deployed in hostile environments. Because of the strict constraints in power, processing and storage, designing an efficient key establishment protocol is not a trivial task. Also, it is infeasible to apply public key techniques onto large-scale wireless sensor networks. Most of proposed solutions are based on symmetric key techniques and mainly focused on key predistribution mechanism. In this paper, we present a new key predistribution scheme using bivariate polynomial combining with expected deployment knowledge. We show that our approach takes advantage in terms of resilience against node compromised over prior schemes with the same resource requirements.**

I. INTRODUCTION

Sensor networks have numerous applications such as home security monitoring, military reconnaissance, target tracking... [1]. Typical sensor networks normally consist of large number of sensor nodes having limited battery power, data processing, and communicate with each other by short-range radio signal. In almost applications, sensor nodes are often spread out randomly over specific regions to sense and collect information.

One of the most basic security requirements for sensor networks is to guarantee the confidentiality and integrity in sending messages between sensor nodes. Environments in which sensor networks are exploited are regularly hostile areas. In these spaces, attackers could eavesdrop on messages or disable the networks by launching physical attacks to sensor nodes, or even using logical attacks to different communication protocols [2], [3]. Thus, to get rid of above problems, sensor networks need encryption and authentication services. Due to resource constraints, implementation an efficient key establishment mechanism is not a trivial task. Since it is impractical to apply public key techniques, all proposals have been using symmetric key techniques so far.

The random key predistribution was firstly proposed by Eschenauer and Gligor [4]. Chan et al. improved with q-composite and random pairwise key predistribution [5]. Du et al. applied deployment knowledge to basic random pairwise key in their scheme [8]. Polynomial-based proposals relied on Blundo's approach [10] are in [11], [12], [13]. The key matrix schemes, developed from Blom's solution [6], are multiple-space key predistribution scheme [7] and DHDV-D [9] of Du et al. All these schemes, although some exploited prior deployment knowledge, still didn't take advantage of this information.

In this paper, we introduce a novel location-aware polynomial-based key predistribution approach in order to improve the security and performance questions. With the advantages of predeployment knowledge, we distribute polynomial information to a limited number of sensor nodes over specific area. So it will decrease the probability to reveal a polynomial when the adversary compromised some nodes. Our scheme is shown to have better security than basic random key preditribution [4], *q*-composite [5] and Closet Polynomials Predistribution Scheme [12], [13].

The rest of the paper is organized as follows: In Section II, we briefly describe related work. Next, Section III gives an overview of Blundo's polynomial key predistribution technique. Section IV presents our proposal in detail. Afterward, we show the analysis and estimation of our scheme compared with others in Section V. Finally, in Section VI, we conclude the paper and point out further research directions.

II. RELATED WORK

The first scheme is proposed by Eschenauer and Gligor [4]. In this system, a large key pool is generated off-line and each sensor picks a random subset of keys from the key pool, called a key-ring. Any two nodes in the communication range can talk to each other only if they share a common key. Depending on the size of the key pool and the number of sensor nodes in the network, this design may achieve different connectivity and resilience. Chan et al. [5] later proposed an approach using the similar idea, but increased the intersection sharing keys between key-rings from one key to some q>1 keys. It is shown that, by increasing the value of q, network resilience against node capture is improved. Du et al. suggested a key predistribution model by applying deployment knowledge [8]. In their design, entire network was divided into groups. Each group implements the basic

[*] Corresponding author.

1-4244-1230-7/07/$25.00 © 2007 IEEE

random key predistribution as in [4]. The key pool of a group shared α keys with horizontal groups' key-pools and β keys with diagonal groups' key-pools.

The key-matrix solutions are based on the idea of Blom [6]. He recommended a key predistribution scheme making certain that any pair of members in a group is able to calculate the common sharing key. Denote *N* is the number of sensor nodes in the network, let *G* be a generator matrix of size *(t+1)×N* over finite field F_q and let *D* be a secret random matrix *(t+1)×(t+1)* with elements in F_q. From the matrix *G* and *D*, construct a *N*×*N* symmetric $K = (D \cdot G)^T \cdot G$ whose entries will be the pairwise keys between nodes. Each node *i* stores a corresponding row *i* of private matrix $A = (D \cdot G)^T$. If node *i* wants to communicate with node j, it computes the inner product of row vector it stores with the *j*-th column of G to obtain the common key $K_{i,j}$. Multiple-space key predistribution of Du et al. [7] combined the Blom's method with the basic random key predistribution of Eschenauer and Gligor [4] for applying to sensor networks. In this approach, they denoted the set of keys that each tuple <D,G> can generate a key space. Each node in the network stored randomly τ spaces from ω pre-generated spaces. Any two nodes could probabilistically share a common space, which may be used to compute a common secret key. Later, Du et al. also applied pre-deployment knowledge to propose DDHV-D scheme in [9]. It is the combination of multiple-space key predistribution [7] with the random predistribution scheme applied deployment knowledge [8]. All the key-matrix solutions have threshold *t*-secure property. It means that if no more than *t* nodes are compromised by attackers then the communications between non-compromised nodes are still secured.

The basic idea of polynomial key generation was proposed by Blundo et al. [10]. It uses symmetric polynomial evaluations to obtain a pairwise key. The detail of this method will be described in the next section. This proposal is *t*-collusion resistant against node captured with property: compromise of less than *t+1* node doesn't reveal any information about keys of other nodes. Derived from above method and basic random key predistribution [4], Liu and Ning introduced random subset assignment key predistribution model [11]. Instead of generating large key-pools and creating key-rings, this scheme creates a large polynomials pool and assigned each node a subset of polynomials from the pool. Then two nodes can only communicate to each other when they shared at least one common polynomial. It is shown that this solution increased the resilience comparing with Eschenauer and Gligor's model [4]. Further solution using predeployment knowledge is Closet Polynomials Predistribution Scheme (CPPS) of Liu and Ning [12], [13]. In CPPS, the entire network is partitioned into rectangular cells. Sensors in each cell store 5 t-degree bivariate polynomials, including the primary polynomial of their cell and polynomials of 4 horizontal neighbors. Then any two nodes sharing at least a polynomial could establish a unique pairwise key. Most of above solutions indicated the trade-off between security and performance.

III. Blundo's Key Predistribution Scheme

Blundo's scheme in [10] uses *n* variables polynomials with *t*-degree to establish key distribution for *t*-secure *n*-conference. Applied to pairwise key between two entities, key predistribution server randomly generates a bivariate *t*-degree polynomial $f(x,y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j$ over a finite field F_q , where q is a large enough prime number that could accommodate a cryptographic key. The function $f(x,y)$ is symmetric, $f(x,y) = f(y,x)$. Each node having unique integer ID i loads the information of $f(i,y)$ from the polynomial $f(x,y)$. Then any two nodes i and j can compute the key $k_{i,j} = f(i,j)$ at node i and $k_{j,i} = f(j,i)$ at node j . Because of symmetric property, we have $k_{i,j} = k_{j,i}$ so that two nodes have a common pairwise key.

Each node must store *t+1* coefficients, each coefficient costs $\log_2 q$ bits. So the memory storage requirement for each node in this model is $(t+1)\log_2 q$ bits. The analysis in [10] shows that, this scheme is unconditionally secure and *t*-collusion resistant. It means that as long as no more than *t* nodes are compromised, the attacker knows nothing about the pairwise key between any two non-compromised nodes.

This basic proposal cannot be applied directly to sensor networks due to its memory overhead for storing keys. The size of memory depends exponentially on the size of the network, so it is not useful for such resource-constraint devices like sensor nodes using only this model. We will focus on this problem by using predeployment knowledge and showing that it will take more advantages than other polynomial-based schemes applying expected location knowledge.

IV. Proposed Key Predistribution Scheme

Before presenting our proposed scheme, we define a key-space as a set of all keys that a *t*-degree bivariate polynomial $f(x,y)$ in Blundo's model could generated. The number of keys in a key-space is denoted as key-space size. We assume that a node will pick a key-space if it carries the information generated from $f(x,y)$. Any two nodes picking a common key-space always compute their pairwise key.

Our scheme has totally three phases: key predistribution, direct key establishment, and indirect key establishment. The key predistribution phase is carried out to preload the credential information to each sensor node before deployment. After set up, two sensor nodes can establish a direct key between them if they share the common key-space. Otherwise, the two sensor nodes could establish a path key by other intermediate nodes' support. At first, we are going to define the deployment model of sensor networks.

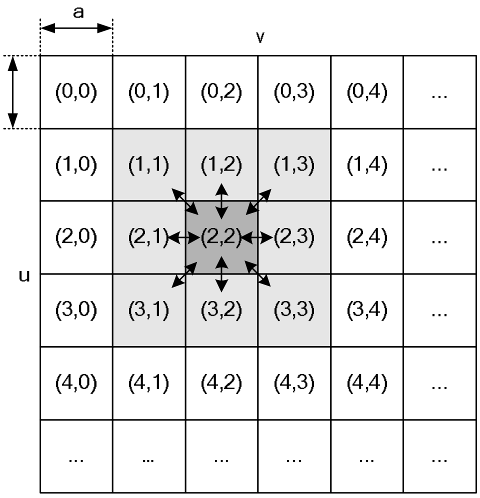
A. Deployment Model

In our proposal, the target field is divided into square-grid with size a×a, for example the one shown in Fig. 1. All sensor nodes that locate in a specific square area have the same cell. This model is practical in realistic, when sensor nodes in each group are spread together, such as using airplane to drop out these groups in sequence, so expected adjacent groups have better chance of being close to each other on the ground. Normally, the arrangement of sensor nodes relies on some probability distribution function. In this case we assume that sensor nodes are uniformly deployed. So each cell has in average N_c nodes.

A sensor node A can receive a message from another sensor node B if A is located within the radio range of B. For simplicity, we model the radio range of a sensor node as a circle centered at node location with radius *R*.

Assume that there are *m* nodes on average in the radio range *R* of each sensor node. So on average number of sensors in the circle area with radius *R* is *m+1* nodes. The density of the sensor nodes in the network can be estimated by $\varpi = \frac{(m+1)}{\pi R^2}$ and the number of nodes in a cell is:

$$N_c = \varpi a^2 = \frac{(m+1)a^2}{\pi R^2}.$$



(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	...
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	...
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	...
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)	...
(4,0)	(4,1)	(4,2)	(4,3)	(4,4)	...
...	...	...	...	...	...

Fig. 1. A square-grid deployment model.

B. Key Predistribution Phase

In this phase, we need to assign key information to each node. After deployment, neighboring nodes can compute a pairwise key between themselves.

We define a cell (*i,j*) is a neighbor of cell (*u,v*) when $|i-u| \le 1$ and $|j-v| \le 1$. It means that each cell has 8 neighboring cells.

With cell (*u,v*) and each of its neighboring cells, key setup server generates a bivariate polynomial *t*-degree and distributes this polynomial to sensor nodes at these two cells. For convenience, each polynomial is assigned a unique ID, denoted as $f_{(u,v),(i,j)}$. It turns out that this polynomial is distributed to sensor nodes at cell (*u,v*) and cell (*i,j*). It is easy to see that $f_{(u,v),(i,j)} = f_{(i,j),(u,v)}$. So every sensor nodes must store knowledge of 8 *t*-degree bivariate polynomials. In other words, each node needs to pick 8 key-spaces information.

C. Direct Key Establishment Phase

After set up, each node must discover whether it shares certain space with its neighbors. To do this, each node broadcasts a message containing the following information: (i) the node's ID, (ii) the IDs of key-spaces it carries.

Suppose that nodes A and B are neighbors, with ID are N_a and N_b respectively. They receive the above broadcast messages from each other. If they find out a common key-space sharing f_c , they could compute the pairwise key as shown in Blundo's scheme: Node N_a computes they key $K_{A,B} = f_c(N_a, N_b)$. Node N_b computes the key $K_{B,A} = f_c(N_b, N_a)$. Because of symmetric property of bivariate polynomial f_c , we have $K_{A,B} = K_{B,A}$. This key is used as the secret pairwise key between node A and node B.

D. Indirect Key Establishment Phase

It may be the case that two sensors *u* and *v* do not share any key-space, they can establish a session key. The source node *u* broadcasts the ID of destination node *v*. An intermediate node *i* receives this broadcast and checks whether it shares the pairwise key with *v* to establish the session key between *u* and *v*. The computation of session key could be performed as following: node *i* plays the role of Key Distribution Center. Then, it computes a session key $k_{u,v}$ and sends this key to node *u* and *v* in encrypted messages by equivalent pairwise keys. If the node *v* is not in the list of neighbor nodes of *i*, node *i* continues broadcasting the message until it reaches to a node that shares pairwise key with *v*. In order to restrict the broadcast storm, the number of hops in the broadcast message should be limited.

E. Sensor Addition and Revocation

To add a new sensor, the key setup server only needs to predistribute the related polynomial shares to the new node, similar to predistribution phase. Since the size of key-space is limited, the more sensors are added, the lower the security in that cell becomes.

The revocation method is also straightforward. Each sensor node only needs to store a black list IDs of compromised sensors that share at least one bivariate polynomial with itself. If there are more than t compromised nodes sharing the same polynomial, the non-compromised nodes that have this polynomial will remove this polynomial and all related compromised nodes.

V. Scheme Analysis

A. Performance Analysis

In this section, we evaluate our scheme with local connectivity, memory and communication overheads.

Based on probability theory, proposals in [4], [5], [7], [8], [9] only guarantee probabilistic key connectivity. It means that they can not provide fully network connectivity. Some parts of the network could be isolated from the rest because of no common key existing. Our scheme is different. When the inequality $a > R$ is assured, a node could establish a pairwise key with any neighbor sensor node, so the local connectivity is guaranteed. The larger the cell size gets, the higher the local connectivity obtains. But when increasing the cell size, the number of sensor nodes sharing a polynomial further increases, so the security decreases. We will discuss more details later.

Applying predistribution knowledge leads to advantages of scalability of network size. About the memory requirement, each node stores 8 key-spaces information, cost $8(t+1)\log_2 q$ bits memory. Beside the key-spaces information, a node also stores its node ID, 8 key-spaces IDs and a black list of compromised nodes. So using our scheme, the memory overhead is not affected when network size changes. The same memory storage only supports limited size of network in solutions of Eschenauer and Gligor [4], Chan et al. [5], but in ours, the network size is unlimited.

In the Direct Key Establishment Phase, the broadcast message only contains the node's ID and 8 IDs of key-spaces. It means that the size of broadcast message is constant when the network resizes.

B. Security against node compromised

The analysis in [10] shows that the polynomial-based scheme has *t*-secure property: unless more than t polynomial shares of a bivariate polynomial are disclosed, an attacker would not know about the non-compromised node's pairwise keys which are established using this polynomial. Thus, the security of our model depends on the average number of sensor nodes sharing the same polynomial, equivalent to the number of sensor nodes that are expected to be located in two neighbor cells.

We have described the deployment model in previous section, the average number of sensor nodes that are expected to be located in a cell is $N_c = \varpi a^2 = \frac{(m+1)a^2}{\pi R^2}$. Thus, the average number of sensor nodes sharing a polynomial can be computed by:

$$N_s = 2N_c = \frac{2(m+1)a^2}{\pi R^2}$$

As long as $N_s \le t$, our scheme is perfect resistance against node captures. In other words, compromising of sensors does not lead to the compromise of direct keys shared between non-compromised sensors.

According to the analysis in [13], we consider a random attack here. We assume a fraction p_c of sensor nodes in the network have been compromised by an attacker. Among N_s sensor nodes that have polynomial shares, the probability that exactly i sensor nodes have been compromised can be evaluated by:

$$P_c(i) = \frac{N_s!}{i!(N_s - i)!} p_c^i (1-p_c)^{N_s - i}$$

So, the probability that the bivariate polynomial is compromised can be calculated by:

$$P_c = 1 - \sum_{i=0}^{t} P_c(i)$$

Fig. 2 demonstrates the relationship between the fraction of compromised direct keys for non-compromised sensor nodes and the fraction of compromised nodes with different combination of m and a. The storage capacity is able to store 200 cryptographic keys meaning that the degree of each polynomial is $t = 24$. Here radio range R is unit distance ($R = 1$). From Fig. 2, we can easily see that, the lower the sensor node density gets, the more security against node captured the scheme achieves. This property is easy to understand because the security against node compromised depends on the number of sensor nodes sharing a key-space. The higher the number of nodes shares, the more vulnerable the key-space gets.

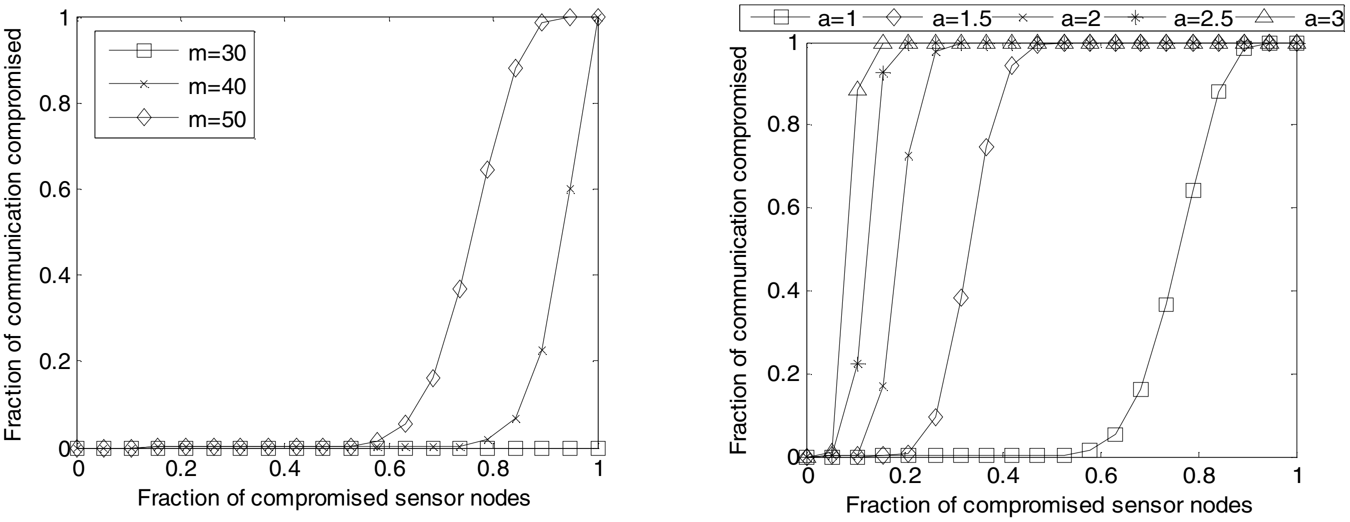
Regarding comparison between our scheme and other schemes: Basic random key predistribution scheme of Eschenauer and Gligor [4], q-composite of Chan et al. [5] with $q = 1$, $q = 3$ and Closet Polynomials Predistribution Scheme (CPPS) of Liu and Ning [12], [13] are shown in Fig. 3. In this scenario, each sensor node could store up to 200 cryptographic keys. The number of neighbor nodes is $m = 40$.

In basic random model, the more compromised nodes it has, the more keys the attacker obtains in the global key pool. So the effect of captured of x nodes by an adversary to communication between uncaptured nodes may be evaluated as in [5]:

$$P_c = 1 - \left(1 - \frac{m}{|S|}\right)^x$$

Where |S| is the key pool size, each node randomly selects a subset *m* keys from the key pool, and *x* is the number of compromised nodes. In this case, the size of key pool is |S| = 100,000 keys, as the same in [4]. With storage $m = 200$, we have the probability of establishing a direct key between two neighbor sensor nodes is $p = 0.33$.

In *q*-composite scheme, the fraction of communications compromised P_c is calculated as in [5]:



(a) a = 1

(b) m = 50

Fig. 2. Network resilience against nodes compromised.

$$p(i)=\frac{\binom{|S|}{i}\binom{|S|-i}{2(m-i)}\binom{2m-i}{m-i}}{\binom{|S|}{m}^{2}};\ p=\sum_{i=q}^{m}p(i)$$

$$P_c=\sum_{i=q}^{m}\left(1-\left(1-\frac{m}{|S|}\right)^{x}\right)^{i}\frac{p(i)}{p}$$

Where |S| is size of key pool, *m* is the size of key subset and *q* is the number of sharing keys between two nodes.

As shown in [5], the probability of any two nodes sharing sufficient keys to form a secure connection is:

$$p_{connect}=1-\sum_{i=0}^{q-1}p(i)$$

Given key ring size *m* = 200, minimum key overlap *q*, and minimum connection probability p = 0.33 as the same in [5], we choose the size of key pool |S| largest such that $p_{connect}\geq p$.

The evaluation of basic random scheme and q-composite requires the network size. Presumably, there are *m* sensor nodes that averagely fall into each sensor's radio range. Based on the analysis of Chan et al. [5], we estimate that the total number of sensor nodes in the network must be $N=2^{mp}$ to make sure the network is fully connected with a high probability if a node only contacts its neighbor nodes, where p is the probability of establishing direct keys between two neighbor sensor nodes.

First, we compare our proposal with basic random key predistribution scheme [4] and the *q*-composite scheme [5]. Fig. 3 shows the fraction of communications compromised between number of compromised nodes given the same p, m and memory overhead. We can see that our scheme is much better security than the other two schemes. It points out the advantage of sensor deployment knowledge affected security level.

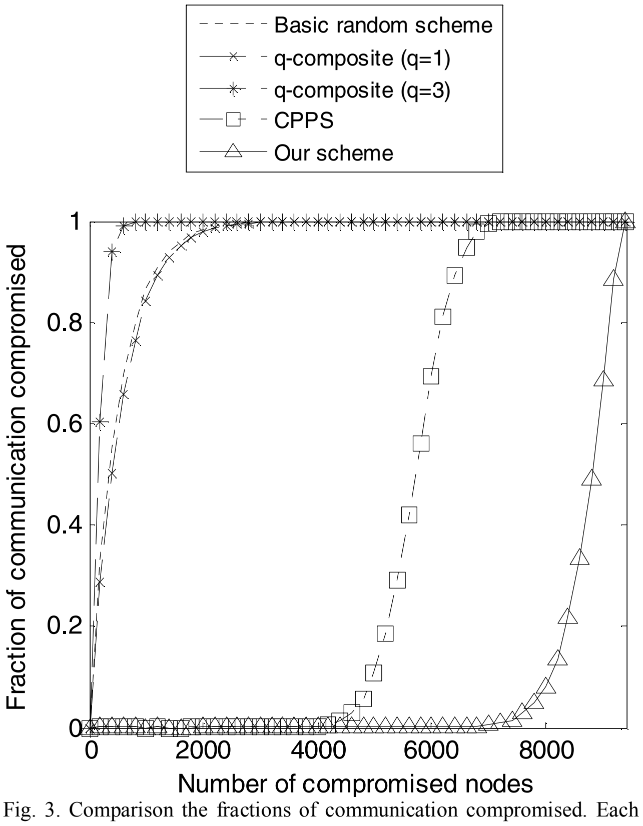


Fig. 3. Comparison the fractions of communication compromised. Each node has storage 200 cryptographic keys. Assume *m*=40 and *p*=0.33.

Next, we compare our model with a location-aware key predistribution scheme, the Closet Polynomials Predistribution Scheme (CPPS) proposed by Liu and Ning [12], [13]. Using the same amount of memory, our model also has better security against node compromised attack. In this scenario, the storage capacity is 200 cryptographic keys. With CPPS, the memory overhead is $5(t+1)\log_2 q$, so the degree of polynomials is t = 39. With our scheme, the memory overhead is $8(t+1)\log_2 q$, so the degree of polynomials is t = 24. Although the degree of polynomials is smaller than CPPS, polynomials in our scheme are distributed to limited number of sensor nodes, so in general, our model gains better resilience against node compromised.

VI. CONCLUSION

In this paper, we have described a polynomial-based key predistribution approach which take advantage of knowledge regarding expected location of sensor nodes. The pairwise keys in the setup phase are computed from the sharing key-spaces between each two nodes. We have shown that this model has more advantages than other schemes in resilience against node compromised attack, more efficiently than others in terms of memory and communication overheads to support large scale network. Our future work will focus more details on performance analysis, with various sensor node distribution models and concerns of error rate in deployment that is the difference between expected location and actual location of sensor nodes.

ACKNOWLEDGMENT

This work is financially supported by the Ministry of Education and Human Resources Development (MOE), the Ministry of Commerce, Industry and Energy (MOCIE) and the Ministry of Labor (MOLAB) through the fostering project of the Lab of Excellency.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci. “A survey on sensor networks,” In *IEEE Commun. Mag.* Vol 40, Issue 8, August 2002.

[2] A.D. Wood, and J.A. Stankovic, “Denial of service in sensor networks,” In *IEEE Computer*. 54-62, October 2002.

[3] C. Karlof, D. Wagner. “Secure routing in wireless sensor networks: attacks and countermeasures,” In *First IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

[4] L. Eschenauer, V. D. Gligor. "A key-management scheme for distributed sensor networks," In *Proceedings of the 9th ACM conference on Computer and communications security*, November 2002.

[5] H. Chan, A. Perrig, D. Song. "Random key predistribution schemes for sensor networks," In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, May 2003.

[6] R. Blom, “An optimal class of symmetric key generation systems,” In Proc of EUROCRYPT ’84, pp. 334-338, 1985.

[7] W. Du, J. Deng, Y. S. Han, P. K. Varshney. "A pairwise key predistribution scheme for wireless sensor networks," In *Proceedings of the 10th ACM conference on Computer and communications security*, 2003.

[8] W. Du; J. Deng; Y. S. Han; S. Chen; P.K Varshney. ”A key management scheme for wireless sensor networks using deployment knowledge,” In *23rd Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom’04)*, Hong Kong, China, March 21-25, 2004.

[9] W. Du, J. Deng, Y. S. Han, P. Varshney. "A Key Predistribution Scheme for Sensor Networks Using Deployment Knowledge," In *IEEE Transactions on Dependable and Secure Computing*, Volume 3 , Issue 1 (January 2006).

[10] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung. ”Perfect-secure key distribution of dynamic conferences,” In *Advances in Cryptography* – CRYPTO ’92, LNCS 740, pp. 471-486, 1993.

[11] D. Liu, P.Ning. "Establishing pairwise keys in distributed sensor networks," In *ACM Transactions on Information and System Security*, February 2003.

[12] D. Liu; P. Ning. “Location-based pairwise key establishments for static sensor networks,” In *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks (SASN '03)*, October 2003.

[13] D. Liu, P.Ning. "Improving Key Predistribution with Deployment Knowledge in Static Sensor Networks," In *ACM Transactions on Sensor Networks*, Vol. 1, No. 2, November 2005, pp. 204-239