Enhanced Group-based Key Management Scheme for Wireless Sensor Networks using Deployment Knowledge

Ngo Trong Canh, P.T.H. Truc, Tran Hoang Hai,
Le Xuan Hung,Young-Koo Lee and Sungyoung Lee
Department of Computer Engineering
Kyung Hee University, Korea
ntcanh@oslab.khu.ac.kr, pthtruc@oslab.khu.ac.kr, haith@icns.khu.ac.kr,
lxhung@oslab.khu.ac.kr, yklee@khu.ac.kr, sylee@oslab.khu.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. Compared with public key cryptography, symmetric key cryptographic with key predistribution mechanism is more suitable for large-scale wireless sensor networks. Most of previous solutions have some issues on performance and security capabilities. In this paper, we propose a novel key predistribution model using pre-deployment knowledge and random values in pairwise key generation to take advantage in terms of network connectivity, memory cost, energy for transmission and strong resilience against node capture attacks.**

***Keywords*: Key predistribution, network security, wireless sensor networks.**

I. Introduction

Sensor networks have a numerous applications such as home security monitoring, military reconnaissance, target tracking [1]. Typical sensor networks normally consist of large number of small devices. Such devices are sensor nodes, having limited battery power, data processing and often communicate with each others 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 wireless 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]. Thus, to get rid of above problems, sensor networks need encryption and authentication services. Due to resource constraints and large-scale network size, implementation an efficient key establishment mechanism is not a trivial task. Beside advantage of elliptic curve cryptography recently, symmetric key algorithms are the feasible solutions to solve this problem, especially for large-scale wireless sensor networks.

The random key predistribution was first proposed by Eschenauer and Gligor [3]. Chan et al. [4] improved with q-composite and random pairwise key predistribution. Du et al. applied deployment knowledge to basic random pairwise key in their scheme [5]. Polynomial-based proposals relied on Blundo's approach [6] are in [7], [8], [9]. The key matrix schemes, developed from Blom's solution [10], are multiple-space key predistribution scheme [11] and DHDV-D [12] of Du et al., hexagonal-based deployment with key matrix of Yu and Guan [13]. All these schemes, although some exploited prior deployment knowledge, still didn't take advantage of this information and have some limited against passive attacks to networks.

We have already proposed a group-based key management model using predeployment knowledge [14]. This approach takes advantage of hexagonal grid and expected location information not only to reduce the memory cost but also get better resilience against captured node attack. But like almost previous models, ours in [14] also have some weaknesses with impacts of node capture attacks. In this paper, we propose some improvements for [14] to eliminate the impacts of node capture attacks on links between non-captured nodes. Our improvements could also apply to similar models as well.

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

II. Blundo's key predistribution model

Blundo's scheme in [6] uses n symmetric 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 symmetric bivariate *t-degree* polynomial $f_(x,y) = \sum_{i,j=0}^{t} a_{i,j}x^i y^j$ over a

978-1-4244-2309-5/09/$25.00 ©2009 IEEE

Authorized licensed use limited to: KYUNGHEE UNIVERSITY. Downloaded on April 13, 2009 at 01:09 from IEEE Xplore. Restrictions apply.

finite field F_q, where q is a large enough prime number that could accommodate a cryptographic key. Each node having unique integer ID i loads the coefficients of $f(i, y)$ derived from $f(x, y)$. 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 [6] 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 would knows nothing about other pairwise keys between any two non-captured nodes.

This basic proposal is not able to apply 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 applied expected location knowledge. Also, we include some random number in evaluating pairwise key, so that it's harder to compromise addition secure links of non-captured nodes.

III. Hexagonal group-based deployment model

In our proposal, the target area is divided in hexagonal grid. This geometry provides the best approximation to circle and covers the biggest area than other two in three geometries that can be repeated over a continuous field: triangle, rectangle and hexagon. Also, a hexagon has the least (six) neighbor cells comparing to eight for rectangle or twelve for triangle. Sensor nodes are partitioned and distributed into groups on cells. This model is practical in realistic scenarios, when sensor nodes in each group are delivered together, such as using aircraft to drop groups in sequence, so expected adjacent groups have better chance of being close to each other on the ground.

Based on different deployment methods, the deployment distributions follows some specific probability distribution functions (pdf). The pdf may be uniform distributions, as in [8], [15] or two-dimensional Gaussian distribution in [11], [5], [12]. In this paper, for the sake of simplify in analysis, we use Gaussian distribution, which is also widely studied and used in practice. Other distributions could be applied as well. Let's assume a sensor networks contains N nodes, which is splitted into G groups, each group is distributed following Gaussian distribution. When the deployment point of group G_i is at (x_i^o, y_i^o), the pdf for a node n_i belongs to group G_i is the following:

$$\begin{aligned} f(n_i(x_i, y_i)|n_i \in G_i) &= \frac{1}{2\pi\sigma^2} e^{-[(x_i - x_i^o)^2 + (y_i - y_i^o)^2]/2\sigma^2} \\ &= f(x_i - x_i^o, y_i - y_j^o), \end{aligned} \quad (1)$$

where (x_i, y_i) is the coordinate of node n_i in the group G_i and σ is the standard deviation of distribution. The value of σ depends mainly on the height of aircraft when dropping sensor groups. We define a cluster is a set of three adjacent groups,

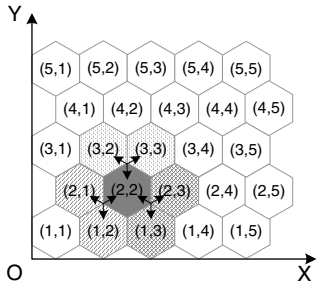

Fig. 1. Hexagonal group-based deployment model.

and there are three types of cluster. At any group $G_{i,j}$, there are 1-cluster containing $G_{i,j-1}$ and $G_{i-1,j}$, 2-cluster containing $G_{i,j+1}$ and $G_{i-1,j+1}$, and 3-cluster containing $G_{i+1,j}$ and $G_{i+1,j+1}$.

IV. Improving Group-based Key Predistribution Using Hexagonal Grid

Before representing our proposal, we define a key-space as a derive from a bivariate polynomial in Blundo's scheme. A node N_A picks a key-space $f_{u,v}(x, y)$ if it carries the coefficients of $f_{u,v}(N_A \oplus nonce_A, y)$, where $nonce_A$ is a random value of node A and will be described in Key Predistribution Phase. When two nodes are in the same key-space, they could calculate a pairwise key to setup a secure channel.

Our scheme allows sensor nodes to find a common key-space with each of their neighbors after deployment. It 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 setting up, two sensor nodes can establish a direct key between them if they share at least a common key-space, otherwise, they could agree on an indirect key based on Indirect Key Establishment phase.

A. Key Predistribution Phase

The purpose of this phase is to assign key materials to each node. Based on these key materials, neighboring nodes could setup pairwise keys after deployment.

This task is done by an offline server. At first, the server will generate a polynomial pool $\mathcal{F}$ containing enough *t-degree* symmetric bivariate polynomials for every clusters. Then it distributes each polynomial to all sensor nodes in each cluster. Because each cell belongs to three clusters, every node has to store knowledge of three *t-degree* bivariate polynomials. In other words, each node needs to pick three key-spaces. The detail algorithm for polynomials predistribution is show on Algorithm 1.

After finishing this phase, every sensor node stores a node-id, three space-id, a random value and three vectors of coefficients equivalent to three key-spaces. These key materials will be used to setup pairwise keys in the next phase.

Authorized licensed use limited to: KYUNGHEE UNIVERSITY. Downloaded on April 13, 2009 at 01:09 from IEEE Xplore. Restrictions apply.

Algorithm 1: Polynomials predistribution

Data: set of nodes $\mathcal{N}$ in the network, set of clusters Ψ, polynomial pool $\mathcal{F}$

Result: Preload key materials to every sensor nodes in the networks

foreach *sensor node* N_A *in* $\mathcal{N}$ **do**
  Generate and insert to preloaded data of N_A a random value $nonce_A$;
end
foreach *cluster* C_i *in* Ψ **do**
  Get a bivariate polynomial $f_i(x, y)$ from $\mathcal{F}$;
  foreach *node* N_A *in* C_i **do**
    Compute $f_i(N_A \oplus nonce_A, y) = \sum_{j=0}^{t} b_j y^j$;
    Insert to preloaded data of N_A: $\{b_j \| j = 0, .., t\}$;
    Insert id of this polynomial, called space-id f_i to N_A,
  end
  Remove $f_i(x, y)$ from $\mathcal{F}$;
end

B. Direct Key Establishment Phase

After deploying to target area, every sensor node must discover the sharing key-space with its neighbors. Assume that node N_A with three its space-ids f_i, f_j, f_k needs to discover sharing key-space with its neighbors. It broadcasts a 1-hop discovery message Key-space Discovery Message (KSDM) containing the followings:

$$N_A, nonce_A, H(f_i \oplus nonce_A),$$
$$H(f_j \oplus nonce_A), H(f_k \oplus nonce_A)$$

in which *H* is the hashing function and $\oplus$ is the xor operation.

When a neighbor of A, let's call B, receives this message, it finds out that it could share three, one or no common key-space with A. Similarly, node A also receives B's KSDM message and finds out common key-spaces. If the sharing is at least one common key-space, the pairwise key between B and A is calculated at B as follow:

$$K_{B,A} = f(N_B \oplus nonce_B, N_A \oplus nonce_A). \quad (2)$$

After getting $K_{B,A}$, node B deletes value $nonce_A$ from its memory. The process of computing pairwise key at A is similar. Because of the symmetric property of bivariate polynomials, $K_{A,B} = K_{B,A}$.

Finishing this phase, every node stores a list of pairwise keys with its neighbors, beside the key-space information and a random value in previous phase.

C. Indirect Key Establishment Phase

In case of there are no common key-spaces between two neighboring nodes, it is needed to establish a path key through one or more intermediate nodes. Our solution for this problem is as follow:

After Direct Key Establishment Phase, every node A knows its set of secured neighboring nodes, denoted as S_A. The source node A wants to establish a pairwise key with its neighbor B, but B and A do not share any key-space. In this situation, A generates a session key, called K_S, and find in S_A a node C that have the same group ID with node B or neighboring group ID of group containing node B. Node A then send a message containing K_S encrypted by key K_{AC} to node C. In turn, node C sends to B the session key through secure channel protected by key K_{CB}. Key K_S then is used as pairwise key between two nodes A and B.

After above three phases, every node stores a table containing neighbor IDs and pairwise keys equivalently. The existence of key materials allows sensor networks to be able to add new nodes for replacement later.

V. Theoretical Analysis and Evaluation

We discuss about the following measurements:

- *Network connectivity*: including local connectivity and global connectivity. Local connectivity is the probability a node could connect with neighbor nodes in its transmission range. Global connectivity is the ratio of the number of sensor nodes forming the largest isolated connected component in the final key graph G to the size of the whole network.
- *Communication overhead*: is the energy a node needs to make communication in proposed model.
- *Memory cost*: that is the memory requirement for storing key materials at nodes in our model.
- *Resilience against node captured*: Adversaries usually launch node capture attacks in order to eavesdrop secure channels in the network, or using key materials revealed from captured nodes to perform node replication attacks. In this analysis, we discuss about whether nodes captured attacks could be used for eavesdropping or not.

A. System configuration

We use the setup in Table I for our simulation and numerical analysis.

TABLE I
Simulation setup

Symbol	Value	Description
N	10,000	Number of sensor nodes in the network.
S	1000×1000 (m^2)	Network deployment area.
R	40 (m)	Sensor node communication range.
M	200 (keys)	The memory for storing key materials.
σ	50 (m)	The standard deviation in Gaussian distribution.

In this scenario, we assume nodes deployment follows a two dimensional Gaussian distribution with pdf function in (1).

Authorized licensed use limited to: KYUNGHEE UNIVERSITY. Downloaded on April 13, 2009 at 01:09 from IEEE Xplore. Restrictions apply.

B. Network connectivity

Assume $A(n_i, n_j)$ is the event node n_i is a neighbor of node n_j, $B(n_i, n_j)$ is the event that share at least one common key-space. The local connectivity could be calculate as

$$\begin{aligned} P_{local} &= P(B(n_i, n_j)|A(n_i, n_j)) \\ &= \frac{P(B(n_i, n_j) \cap A(n_i, n_j))}{P(A(n_i, n_j))}. \end{aligned} \quad (3)$$

Probability that a node $n_i \in G_i$ is a neighbor of node $n_j(x_j, y_j)$ is the integral of pdf $f(n_i)$ over the circle around node n_j with radius R

$$P(n_j(x_j, y_j)) = \iint_{G_i, ||(x,y),(x_j,y_j)|| \leq R} f(n_i(x, y))dxdy. \quad (4)$$

Because n_j distribute in group G_j following (1), the probability that $n_i \in G_i$ is a neighbor of $n_j \in G_j$:

$$P(A(n_i, n_j)||G_i, G_j) = \iint_{G_j} P(n_j(x_j, y_j))f(n_j(x, y))dxdy. \quad (5)$$

Hence,

$$\begin{aligned} P(A(n_i, n_j)) = \sum_{G_i \in \Psi} \sum_{G_j \in \Psi} & P(n_i \in G_i)P(n_j \in G_j) \\ & P(A(n_i, n_j)||G_i, G_j). \end{aligned} \quad (6)$$

Denoted $S(G_i)$ is set of neighboring groups of G_i, we have

$$\begin{aligned} P(B(n_i, n_j) \cap A(n_i, n_j)) = \sum_{G_i \in \Psi} \sum_{G_j \in S(G_i)} & P(n_i \in G_i) \\ & P(n_j \in G_j)P(A(n_i, n_j)||G_i, G_j). \end{aligned} \quad (7)$$

Because a sensor node is chosen in a given group with an equal probability, we have the local connectivity can be calculated as

$$P_{local} = \frac{\sum_{G_i \in \Psi} \sum_{G_j \in S(G_i)} P(A(n_i, n_j)||G_i, G_j)}{\sum_{G_i \in \Psi} \sum_{G_j \in \Psi} P(A(n_i, n_j)||G_i, G_j)}. \quad (8)$$

With the high local connectivity, the key sharing graph G may have many isolated components. In this case, the global connectivity could be very low. Because the deployment distribution follows Gaussian distribution, there are 99.87% sensor nodes of a group reside within range from its deployment point. If two adjacent deployment points $d = a \times \sigma$ is far enough, each sensor group forms a isolated component. But if d is small, neighboring groups will overlap each others, so that local connectivity will be low. The global connectivity could be not high, when some sensor nodes, because of d is too small, deliver away from its home cell and six neighboring cells. They could connect with neighbors, but could not setup secure channels with others, hence become orphan sensor nodes. Because the final goal of key establishment in wireless sensor networks is to form networks with as high global connectivity as possible, the distance of adjacent deployment points d must be chosen suitably.

The simulation result in Table II shows the varies of connectivity and the distance of two adjacent deployment points d, which is normalized by the Gaussian standard deviation σ.

TABLE II
SIMULATION RESULT

a	Local connectivity	Global connectivity
0.4	0.0787	0.6546
0.6	0.1577	0.9290
0.8	0.2524	0.9704
1.0	0.3643	0.9921
1.5	0.6036	0.9990
2.0	0.7720	0.9994
2.5	0.8617	0.9998
3.0	0.9226	0.9999
3.5	0.9555	0.9999
4.0	0.9657	1

When the distance between two deployment points of two neighboring cells is too low (a = 0.4; 0.6; 0.8 or 1.0), at any node A, there are many nodes of non-neighbor cells distributed around it. These nodes do not share any key-space with node A. So the local connectivity and global connectivity are reduced. From Table II, it is easy to see that our model gains high local and global connectivity when choosing suitable value of deployment point distances. With value $a = 1.5$, the global connectivity is 0.9990, meaning that only 0.01% number of nodes in the network are waste.

C. Memory cost and Communication overhead

The long lived time is the critical goal in designing protocols for wireless sensor networks. In our proposal, we minimized the broadcast data requirement in discovery common key-space between neighboring nodes. Our 1-hop broadcast message length is $sizeof(node\_id) + sizeof(nonce) + 3 \times sizeof(hash)$. Comparing with other models in [3], [4], the broadcast messages in key discovery phase contain hundreds of key, to achieve high connectivity.

The memory size for storing key materials derived from polynomials is M = $3 \times (t + 1)log_2 q + sizeof(node\_id) + sizeof(nonce)$ (bits). This value, along with number of nodes sharing a polynomial, affects to the resilience against node compromised attacks before pairwise key established. We will discuss more detail in the following section.

D. Resilience against Node Capture Attacks

The proposed model totally has two stages that could be concerned by node capture attacks. The first stage is the period after distributing sensor nodes to the target field, but before establishing pairwise key. The second is after broadcasting discovery messages and forming a connected pairwise keys network. The properties of two stages lead to different security levels.

In the first stage, adversaries could perform capturing enough sensor nodes in a cluster to interpolate polynomials,

Authorized licensed use limited to: KYUNGHEE UNIVERSITY. Downloaded on April 13, 2009 at 01:09 from IEEE Xplore. Restrictions apply.

then eavesdropping KSDM messages to get $nonce$ values. Number of enough captured sensor nodes depend on the polynomial degree, that is the memory for storing key materials. In this manner, attackers could generate all pairwise keys in the cluster. So that the security in this stage needs some consideration. But we need to remind that the time of this stage is quite short when right after deployed, sensor nodes perform key discovery and establishment, hence the chance for adversaries is not high.

In the second stage, that pairwise keys established, from (2), we can see that each pairwise key depends not only on key-space information but also on random values associated with each node. Assume that a node A is compromised and adversaries could get all information storing inside A's memory. In this case, adversaries could only get pairwise keys between A and its neighbors, a fraction of polynomial $f(N_A, y)$. Even they may captured large enough nodes like A, they definitely could not expose pairwise key between non-captured nodes B and C because K_{BC} require the random values storing inside B and C. So, comparing with other schemes in [3], [4], [11], [7], [8], [5], [12], [15], [14] ours has not been affected by node capture attacks on links between non-compromised nodes.

VI. Conclusion

In this paper, we have described a realistic polynomial-based key predistribution approach which take advantage of predeployment knowledge with Gaussian distribution. We have shown that this model has advantages in network connectivity, communication overhead, memory requirement. It also eliminates the threat of addition keys compromised between non-captured nodes. Our future work will focus on impacts of deployment error rate to network connectivity and analysis performance with multi-hop indirect key establishment.

VII. Acknowledgment

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement)" (IITA-2008-(C1090-0801-0002)) and This research was supported by the MIC(Ministry of Information and Communication), Korea, Under the ITFSIP (IT Foreign Specialist Inviting Program) supervised by the IITA (Institute of Information Technology Advancement: C1012-0801-0003) and 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, and E. Cayirci, "A survey on sensor networks," *Communications Magazine, IEEE*, vol. 40, no. 8, pp. 102–114, August 2002.

[2] A. Wood and J. Stankovic, "Denial of service in sensor networks," *IEEE Computer*, vol. 40, no. 10, pp. 54–62, October 2002.

[3] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pp. 41–47, November 2002.

[4] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," *Security and Privacy, 2003. Proceedings*, pp. 197–213, 11-14 May 2003.

[5] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," *23rd Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom'04)*, pp. 197–213, March 21-25 2004.

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

[7] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 1, 2005.

[8] D. Liu and P. Ning, "Improving key predistribution with deployment knowledge in static sensor networks," *ACM Transactions on Sensor Networks*, vol. 1, no. 2, pp. 204–239, November 2005.

[9] Y. Zhou, Y. Zhang, and Y. Fang, "Llk: a link-layer key establishment scheme for wireless sensor networks," *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 4, pp. 1921–1926, 13-17 March 2005.

[10] R. Blom, "An optimal class of symmetric key generation systems," *Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques*, pp. 334–338, 1985.

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

[12] W. Du, J. Deng, Y. Han, and P. Varshney, "A key predistribution scheme for sensor networks using deployment knowledge," *Dependable and Secure Computing, IEEE Transactions on*, vol. 3, no. 1, pp. 62–77, Jan-March 2006.

[13] Z. Yu and Y. Guan, "A robust group-based key management scheme for wireless sensor networks," *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 4, pp. 1915–1920, 13-17 March 2005.

[14] N. T. Canh, Y.-K. Lee, and S. Y. Lee, "Hgkm - a group-based key management scheme for sensor networks using deployment knowledge," *Sixth Annual Conference on Communication Networks and Services Research (CNSR '08)*, 5-8 May 2008.

[15] N. T. Canh, T. V. Phuong, Y.-K. Lee, S. Y. Lee, and H. Lee, "A location-aware key predistribution scheme for distributed wireless sensor networks," *15th IEEE International Conference on Networks (ICON '07)*, November 2007.

Authorized licensed use limited to: KYUNGHEE UNIVERSITY. Downloaded on April 13, 2009 at 01:09 from IEEE Xplore. Restrictions apply.