On Grid-Based Key Pre-distribution: Toward a Better Connectivity in Wireless Sensor Network*

Abedelaziz Mohaisen**, YoungJae Maeng, and DaeHun Nyang***

Information Security Research Laboratory - INHA University
253 YongHyun-dong, Nam-Gu, Incheon 402-751, Korea
{asm,brendig}@seclab.inha.ac.kr, nyang@inha.ac.kr

Abstract. In this paper, we revisit Grid-Based Key Pre-Distribution Scheme in Wireless Sensor Network to investigate improving the connectivity of the network and maintain both the security level and communication overhead. Both of the original work and our modification are based on using symmetric bivariate polynomials for generating cryptographic keys. In addition, their work relies on the usage of multi-dimensional grid to assign the polynomials on the sensor nodes allocated on the intersections of the grid and provide a needed connectivity. In this work we consider the simplification of the key establishment logic, the enhancement of connectivity in what we call the plat polynomial assignment. We present detailed discussion on the connectivity, resources usage, and security features that shows better results on the side of the connectivity, intermediate node discovery and security measurement. Finally, we provide a comparison between our results and other existing solutions including the revisited scheme.

Keywords: Sensor Networks, Key Distribution, Bivariate Symmetric Polynomials, Network Connectivity.

1 Introduction

Wireless Sensor Network is a resulting successful coalescence of different technologies that includes microelectronics and semiconductors, networking, signal processing, and others [1]. This network consists of large number of sensor nodes which are inexpensive devices with limited resources that work in cooperative method to perform some sensing tasks [2]. Sensors communicate in peer-to-peer fashion in an open air environments that provide opportunities for man-in-the-middle (MITM) [3], Sybil [3, 4], or node replication attacks [5]. The growth of WSN applications brings the necessity to provide security rules to guard the

* This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement)(IITA-2006-C1090-0603-0028).

** Mohaisen's work was done when he was graduate student at INHA University.

*** Corresponding author.

T. Washio et al. (Eds.): PAKDD 2007 Workshops, LNAI 4819, pp. 527–537, 2007.

communication traffic. The symmetric key cryptography which is based on using the same key in both of the two communication entities to encrypt and decrypt messages is very efficient on the typical sensor node's platform [1, 6]. Due to the weak infrastructure of the WSN, the bottleneck challenge in the security is the distribution of keys on sensor nodes [7, 6, 8]. Traditional key distribution mechanisms such like the Trust Third Party (TTP) or the Key Distribution Center (KDC) are impossible solutions as well [9]. Therefore, it is required to be wisely distributed on the nodes in pre-deployment phases. In the next subsection, we provide a survey on the key pre-distribution schemes in WSN followed by our contribution and the structure of this paper.

1.1 Researches on Key Pre-distribution: Survey

General Schemes: Two of the early works in [10, 11] are widely known for its novelty. In the first work by Blom *et al.* [10] a symmetric matrix of size $N \times N$ is required to store the different N^2 keys for securing communication within the entire network of N nodes. Node $s_i \in N$ has row and column in the matrix. If two nodes s_i, s_j would like to communicate, they use the entries $\mathbf{E_{ij}}$ in s_i side and $\mathbf{E_{ji}}$ in s_j side which are equal (*i.e.* $\mathbf{E_{ij}} = \mathbf{E_{ji}}$ since the matrix is symmetric). To reduce the memory requirements, a slight modification is introduced by Du *et al.* [12]. The following are defined, a public matrix $\mathbf{G}$ of size $(\lambda+1) \times N$ and a private symmetric matrix $\mathbf{D}$ of size $(\lambda+1)\times(\lambda+1)$ where $\mathbf{D}$ entries are randomly generated. Also, $\mathbf{A} = (\mathbf{D} \cdot \mathbf{G})^{\mathbf{T}}$ of size $N \times (\lambda + 1)$ is defined. For a node s_i, row $\mathbf{R_i}$ in $\mathbf{A}$ and column $\mathbf{C_i}$ in $\mathbf{G}$ is selected. When two nodes s_i, s_j would eventually communication securely, they exchange their $\mathbf{C_i}, \mathbf{Cj}$ and compute $k_{ij} = \mathbf{R_i} \cdot \mathbf{C_j}$ in the side of s_i and $k_{ji} = \mathbf{R_j} \cdot \mathbf{C_i}$ in the side of s_j. The second work by Blundo *et al.* [11] uses Symmetric Bivariate Polynomial (SBP) to distribute keys for N nodes. The SBP is in the form of $f(x, y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j, (a_{ij} = a_{ji})$ of degree $t \leq N$. For a node s_i with identifier i, the share $g^i(y) = f(i, y)$ is calculated and loaded to its memory generate secure keys. Similarly, for two nodes s_i, s_j, $k_{ij} = g^i(j), k_{ji} = g^j(i)$ are are evaluated locally and used respectively.

Random Key Pre-distribution Schemes: The early scheme of key pre-distribution specifically for WSN is introduced by Eschenauer-Gligor (EG) [13]. Each node is let to randomly pick a key ring S_k of size k from big keys pool of size P. The picking process provides a probabilistic connectivity $p_{actual} = 1 - \frac{((P-k)!)^2}{(P-2k)!P!}$. If two nodes s_i, s_j share a key $k : k \in S_{k_i} \cap S_{k_j}$ they use it a secret key. Otherwise, a path discovery phase via intermediate nodes is performed. In [13], the usage of memory is reduced, however, a weak resiliency is resulted. To improve the resiliency, Chan *et al.* proposed the q-composite scheme [14]. Using the same procedure of EG, a key between two nodes s_i, s_j is available **iff** $S_{k_i} \cap S_{k_j}$ is a set of q keys. If $\{k_1, \ldots, k_q\} \in \{S_{k_i} \cap S_{k_j}\}$, $\mathbf{hash}(\mathbf{k_1}||\mathbf{k_2}, \ldots, ||\mathbf{k_q})$ is used as k_{ij}, k_{ji}. Otherwise, intermediate node(s) are used. More analytical analysis on the probabilistic schemes is shown in by Kwang and Kim in [15].

Symmetric Matrices Based Key Pre-distribution: In addition to improving Blom's scheme in [10], Du *et al.* proposed two schemes [16, 12]. In the early

one they introduced a deployment knowledge based scheme that improves Blom's [10] by avoiding the unnecessary memory, communication, and computation with reasonable connectivity [16]. In [12], a multi-space matrix scheme based on [10, 13] is introduced. A τ number of private matrices $\mathbf{D}$ is selected randomly out of ω pre-constructed matrices providing connectivity of $p_{actual} = 1 - \frac{((\omega-\tau)!)^2}{(\omega-2\tau)!)\omega!}$. Different $\mathbf{A}$'s are created using the different $\mathbf{D}$s. τ rows of the different $\mathbf{A}$s are selected and assigned for each node. For (s_i, s_j), If they have a common space $\tau_{i,j} : \tau_{i,j} \in \tau_i, \tau_j$, the rest of Blom's is performed, else, an intermediate space is used to construct a key path. Even though much memory and communication are required and smaller connectivity is generated, this work provides a higher resiliency than in [13, 14]. For more accuracy, different deployment structures with practical error measurements and the probability distribution functions pdf based on [16] is used by Ito *et al.* in [17].

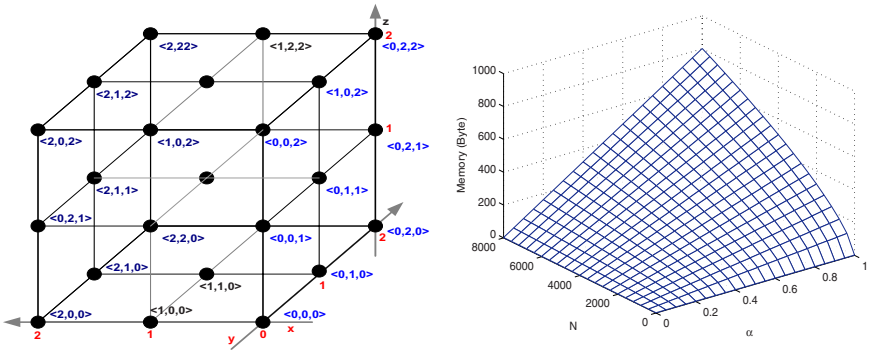
Symmetric Bavariate Polynomial Based Schemes: At the same time, Liu *et al.* proposed several schemes [9, 18] for key distribution and mainly based on [11]. In [19], Blundo's scheme is used by assigning more than polynomial for each node similar as in EG scheme [13]. Another scheme is introduced where for a network of size N a two dimensional deployment environment constructing a grid of $N^{1/2} \times N^{1/2}$ is constructed. The different nodes are deployed on different intersection points of the grid and different polynomials are assigned for the different rows and columns of the grid. For any two nodes s_i, s_j, if $R_i = R_j$ or $C_i = C_j$, (*i.e.* they have the same SBP share), a direct key establishment as in [11] is performed. Else (*i.e.* $R_i \neq R_j$ *and* $C_i \neq C_j$), an intermediate node is used in indirect key establishment phase. Even if a big fraction of nodes p_c as 50% of N is compromised, the network still has the ability to be connected via alternative intermediate nodes. an n-dimensional scheme is introduced in [18].

1.2 Our Contribution and Paper Organization

We improve the three dimensional grid based key pre-distribution scheme in [9, 18] using an extended order of grid with a plat polynomial assignment. Our contribution includes the nodes/polynomials assignment in a three dimensional grid, performance analysis to support the improvement of the connectivity and security performance. We provide a detailed study on the usage of the different resources and study the impact of the communication traffic modeling on required resources. On the structure of the paper, section 2 introduces our detailed contribution, section 3 introduces an analysis of the connectivity, resources usage, performance and security. Section 4 introduces a comparison between our scheme and other possibly comparable schemes and section 5 introduces a conclusion and set of remarks.

2 Grid-Based KPD with Plat Polynomial Assignment

Consider a network of size N, we use a three dimensional grid as in Fig. 1(a). The three dimensional grid has x, y, z axis with each of $N^{1/3}$. We denote the columns



(a) Grid structure (b) Memory requirements

Fig. 1. (a) The structure of the grid with plat-polynomial assignment (b) Memory required to store the polynomial for different network and security parameters

of the different dimensions as C_x, C_y, C_z. The different nodes are deployed on the intersections of the grid. Each node s_i has a three axis coordinate as $\langle C_x, C_y, C_z \rangle$. For each axis of the grid, different symmetric polynomials are assigned in a way that all of the plat with the same axis value owns the same corresponding polynomial. The total number of the polynomials in the network is $3(N^{1/3})$. Each node in the network has three polynomials shares of the corresponding axis. In the following we provide details on the keying material generation and secure key establishment.

2.1 Keying Material Assignment and Identifiers Structure

On a key management server, the following procedure is performed for one time:

- To construct the grid introduced earlier as of Fig. 1(a), the server picks an integer $m = \lceil N^{1/3} \rceil$ where N is a larger integer than the real network size to provide a flexible extendibility for the network size that provides the ability for other nodes to join and leave the network freely.
- Each sensor node s_i with identifier (i) of size $\log_2 N$ bits is mapped to the proper position on the grid of Fig. 1(a) for which the node will have the ID structure of $i = \langle c_x || c_y || c_z \rangle$.
- The server construct $3 \times m$ different symmetric polynomials. Each polynomial satisfies the condition that $f(x, y) = f(y, x)$ where $f(x, y)$'s coefficients are randomly picked in a finite field F_q with $q > N$ to avoid the collision and accumulate a reasonable security stand.
- The different polynomials are grouped in triples that constructs all of the possible outcomes for three groups of m size. Each group also has the notation $< f_{c_x}, f_{c_y}, f_{c_z} >$.
- Unlike [19], each node with the identifier $\langle c_x || c_y || c_z \rangle$ picks three polynomials with the indices (c_x, c_y, c_z) that are equal to the node's identifier coordinates (*i.e.* all nodes with same plat has same polynomial).

– For each sensor node s_i with identifier i and polynomials $< f_{c_x}, f_{c_y}, f_{c_z} >$, the server evaluate the shares $g_{c_x} = f_{c_x}(i, y), g_{c_y} = f_{c_y}(i, y), g_{c_z} = f_{c_z}(i, y)$ and load these shares into s_i's memory.

Note that this will guarantee that all of the nodes with the same x-axis (*i.e.* that belongs to the same plat of the same dimension) have the same polynomial.

2.2 Key Establishment

Assuming that limited number of nodes is compromised, the assigned polynomial for each node provides actual connectivity that enables two different nodes to communicate directly using their polynomials shares. Other cases require using intermediate nodes to establish keys. In the following we show the different cases.

Direct Key Establishment: For two nodes s_i, s_j that has the IDs $i = \langle c_x||c_y||c_z\rangle$, $j = \langle c_x||c_y||c_z\rangle$, if $i.c_x = j.c_x$ or $i.c_y = j.c_y$ or $i.c_z = j.c_z$; that means that both s_i, s_j belong to at least one common dimension and have a shared polynomial share, use the common shared polynomial's share $g^*(y)$ to establish a common key. If $i.c_x = j.c_x$ and $i.c_y = j.c_y$, or $i.c_x = j.c_x$ and $i.c_z = j.c_z$, or $i.c_y = j.c_y$ and $i.c_z = j.c_z$ use the common polynomial with least compromised shares to establish the key (Note that the last case where all of the coordinates are equal is concerned with the node itself). Finally, If neither of the node's coordinates is equal, an intermediate node or more are used to establish a key using the Indirect Key Establishment.

Indirect Key Establishment: If the two nodes don't belong to the same plat of dimension, they should establish a key path that consists of one or more intermediate nodes. For nodes s_i, s_j as communication parties with i, j as identifiers, they pick s_ϕ with ϕ as the identifier such that any of the following is satisfied:

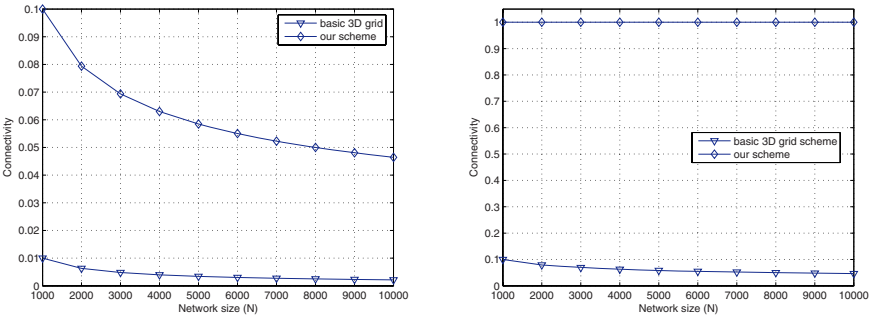
– $\phi.c_x = i.c_x$ and $\phi.c_y = j.c_y$ or $\phi.c_z = j.c_z$.
– $\phi.c_y = i.c_y$ and $\phi.c_x = j.c_x$ or $\phi.c_z = j.c_z$.
– $\phi.c_z = i.c_z$ and $\phi.c_x = j.c_x$ or $\phi.c_y = j.c_y$.

Based on the matching of the parts, the corresponding shares in ϕ are used (at least two) to make the node s_ϕ as an intermediate node. For example, the first matching leading to the following key generations $k_{\phi i} = g^{\phi}_{cx}(i)$, $k_{i\phi} = g^{i}_{cx}(\phi)$, $k_{\phi j} = g^{\phi}_{cx}(j)$, and $k_{j\phi} = g^{j}_{cx}(\phi)$ and so on for other matchings.

3 Analysis

3.1 Distribution of Communication

In the wireless network, the communication activities are distributed and modeled using a communication traffic function with a probability distribution function f_R defined on the area R. To use this advantage, we introduce $f_R(n)$ that is defined on the area and the plats to which nodes are related. In other words, n would mean the number of hop under the non-adversary condition.



(a) Single hop connectivity (b) Two hop connectivity

Fig. 2. The connectivity comparison for our scheme and the grid based scheme [18] for both the single and two hop operation

3.2 Connectivity

The connectivity in general is defined as the fraction of the nodes that can communicate with each other using its own keying material in one hop manner. In our scheme, the directly guaranteed actual connectivity is due to the three plats' ploynomials that represents the axis that the node resides on. Let $m = N^{1/3}$, the connectivity of our scheme is $C_{actual} = 3(\frac{m^2-1}{m^3-1})$ which is approximately equal to $\frac{3}{m+1}$. For the basic grid scheme, the single hope connectivity is $C_{actual} = 3(\frac{m-1}{m^3-1}) = (\frac{3}{m^2+m+1})$ [18], which is smaller than the corresponding in our scheme. Fig. 2 shows a comparison for the connectivity between our modified scheme and the original work where Fig. 2(a) is for the single hop connectivity and Fig. 2(b) is for the two hop connectivity. Note that the connectivity of our scheme is always better than the corresponding of [18] since $\frac{3}{m+1} > \frac{3}{m^2+m+1}$ for any $N > 0$. Practically, the minimum network size under which the grid construction is valid is $N = 8$ which leads to $m = 2$ that guarantees the validity of the connectivity advantage in our scheme.

3.3 Memory Overhead

The memory is required mainly dependent on the desirable security level. Let (α) be a security parameter such that $0 \leq \alpha \leq 1$ which determines the level of the security for the nodes which hold the shares of a given polynomial [15], the required memory for storing the coefficients $a_0, a_1, \ldots, a_t$ of the polynomial terms $x^0, x^1, \ldots, x^t$ is $(t+1)\log_2(q)$ bits which can be written as $(\alpha \times m + 1) \times \log_2(q)$. Let N_c be the number of compromised nodes, the required memory M is as in Eq. 1. The required memory for different parameters is shown in Fig. 1(b)

$$M = 3\Big((N_c + 1)\left\lceil \log_2(N^{1/3})\right\rceil + (t+1)\log_2(q)\Big) \tag{1}$$

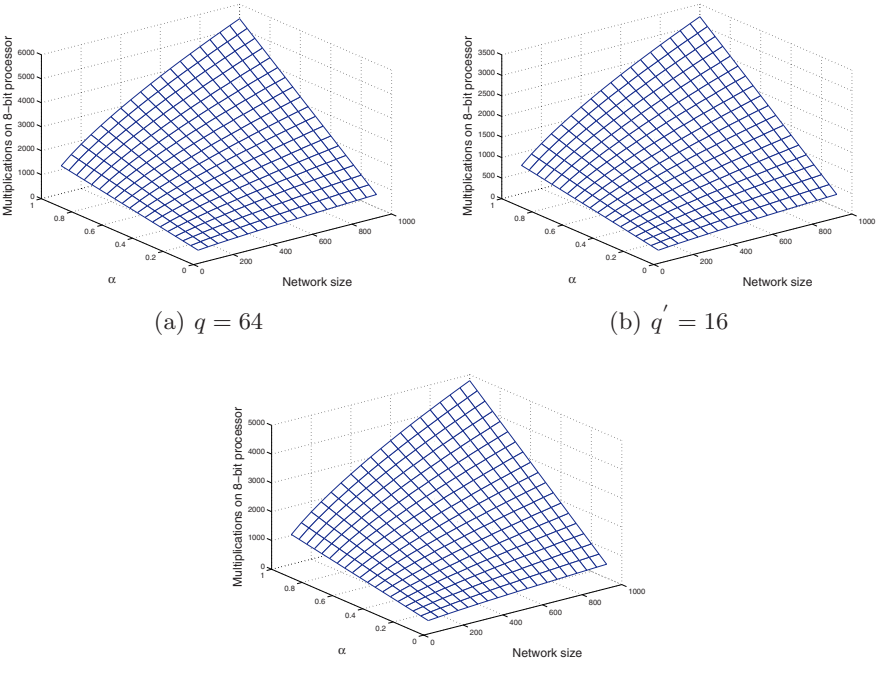
3.4 Communication Overhead

The security-related communication overhead is required to exchange two nodes identifiers. For the key establishment under the condition that small fraction of nodes is compromised, there are two different cases: The direct key establishment that requires single ID exchange and the KPE through one intermediate node that requires two identifiers exchange. Based on the identifier structure presented earlier, $3\log_2(N^{1/3})$ bits are required to represent it. At average, the required communication overhead in bits is the average to exchange the IDs in the two cases as follows:

$$C_{cm_{avg1}} = \frac{1+2}{2} \times 3 \left\lceil \log_2(\sqrt[3]{N}) \right\rceil = 4.5 \left\lceil \log_2(\sqrt[3]{N}) \right\rceil \tag{2}$$

For practical models that consider the usage of the communication traffic function, the communication overhead is defined as:

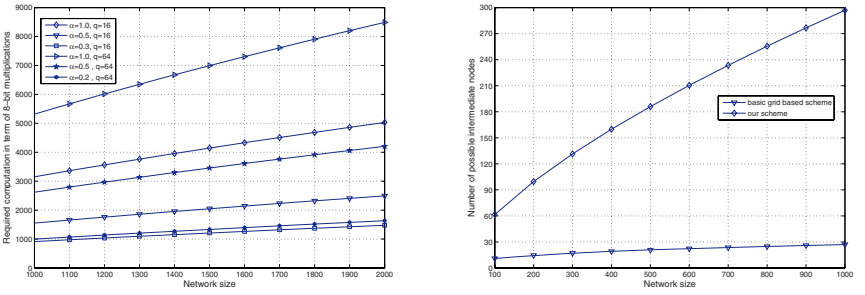
$$C_{cm_{avg2}} = 3 \left\lceil \log_2(\sqrt[3]{N}) \right\rceil \sum_{i=1}^{n} i f_R(i) \tag{3}$$



(a) $q = 64$ (b) $q^{'} = 16$

(c) $q^{'} = 16$ without $f(n)$

Fig. 3. Computation overhead for network and security parameters: varying q, α, n with and without communication traffic function $f(n)$ where $f(n) = \frac{c}{2^{n-1}}$



(a) Computation overhead (b) Number of intermediate nodes

Fig. 4. (a) The computation overhead in term of required multiplications on 8-bit word processor using different security parameters (b) Possible intermediate nodes for two hops secure communication in our scheme and in the basic grid-based scheme

3.5 Computation Overhead

The same like in the communication, the computation has two different case: one polynomial evaluation of degree t is required if the two nodes are in the same plat. Else, two or more polynomials evaluation is required. For the general case, $f_R(n)$ is used to determine the required average computation within the running time of the network life. The first case's computation requirement is $C_m = 2t - 3$ multiplications on a big integer field to evaluate a polynomial of degree $t = \alpha \times m^2$ where $m = N^{1/3}$ and the second case's computation requirement is shown in the following in Eq. 4.

$$C_c = C_m \sum_{i=1}^{n} i f_R(i) \tag{4}$$

Based on [19], two integers' multiplication on a finite field of 16 or 64 bits requires 16 or 27 8-bit multiplications respectively. Fig. 3 and Fig. 4(a) show comparisons between the required computations on both the 16 an 64 finite fields for different security parameters. Fig. 3(a) considers $q = 64$ and Fig. 3(b) considers $q^{'} = 16$ with $F_R(n)$. Fig. 3(c) considers the computation without $F_R(n)$ which is notably greater than the early one.

3.6 Security Analysis

The security of any polynomial-based scheme is driven from the fact that the polynomial is secure as long as the number of compromised nodes is less than $t + 1$. In the following we measure different situations of our scheme:

– **Compromise of single node:** The single node holds one share for a concerned polynomial, even if it holds two other shares for two different polynomials. Thus, the compromise of single node will not reveal more than the share of the sensor node and other internal information.

– **Compromise of single plat:** We define the single plat by the nodes which holds the same polynomial's shares which are required to recover a polynomial when $\alpha = 1$. The probability for this event to happen when the number of the compromised nodes N_c is given is p_c as shown in Eq. 5.

$$p_c = 1 - \sum_{i=0}^{t} \left(\frac{(N^{\frac{2}{3}})!}{i!(N^{\frac{2}{3}} - i)!} \right) \left(F_c \right)^i \left(1 - F_c \right)^{N^{\frac{2}{3}} - i} \tag{5}$$

where F_c is the fraction of compromised nodes, i is the number of compromised shares of a given polynomial, and N is the network size. An example that illustrates the difference between our scheme and the scheme in [18], for $N = 1000$ and $F_c = 0.5$, $p_c \approx 0.2$ in our scheme while $p_c \approx 0.4$ for the same network size for the construction of [18].

– **Attack against the network:** The attack against the whole network to break the security of the pair-wise communication between the sensor nodes is possible by compromising every single polynomial using the above approach. Even though there is a big overlap between the nodes which hold shares for different polynomials, still the network can resist up to a big fraction of compromised nodes and the revealed shares will be useless till a determined threshold fraction determined by α.

– **Possible intermediate nodes for key path establishment:** Unlike the basic grid scheme in [18] which provides $3\sqrt[3]{N}$ possible intermediate nodes for any key path establishment through two hops, our scheme provides $3\sqrt[3]{N}^2$ for the same process under the assumption of limited nodes are compromised. Fig. 4(b) shows a comparison between our scheme and [18].

4 Comparison with Other Schemes

The security of our scheme is typically same like the original work of Liu-Ning's [9]. The comparison between our scheme and other schemes is defined in the

Table 1. Comparison between our scheme and a set of other schemes in term of the resources usage and the resulting connectivity. The connectivity for the probabilistic key pre-distribution schemes is probabilistic while it is certain for other schemes including ours. Also, the polynomial degree t differs as shown earlier based on α.

Scheme	Communication	Computation	Memory	Connectivity
GBS [18]	c	SBP Evaluation	ID+2 SBP	$\frac{2}{N^{1/2}-1}$
3D-GBS [18]	c	SBP Evaluation	ID+3 SBP	$\frac{3}{N^{2/3}+N^{1/3}+1}$
our scheme	c	SBP Evaluation	ID+3 SBP	$\frac{\mathbf{3}}{\mathbf{N^{1/3}}}$
EG [13]	$c \log_2(S_k)$	$\frac{(2C+p-p_k)}{2} \log_2(C)$	S_k keys	$1 - \frac{((P-k)!)^2}{(P-2k)!P!}$
CPS [14]	c	c	S_k keys	$\frac{m}{N}$
DDHV [12]	$c \log_2(n \times \tau)$	2 vectors mult.	$\tau + 1$ vectors	$1 - \frac{((\omega-\tau)!)^2}{(\omega-2\tau)!\omega!}$
HGBS[20]	c	SBP Evaluation	ID+n SBP	1

resources usage shown in section 3. Table 1 shows a list of resources comparison with other possible schemes for key distribution in wireless sensor networks. This includes grid and three dimensional grid based schemes [9], hierarchical grid scheme. [20], Du *et al.* [12], and our scheme. The comparison shows that relatively, the usage of resources is relatively comparable with the corresponding in [9, 18]. As well, considering the advantageous connectivity shown earlier.

5 Conclusions

In this paper, we revisited the grid-based key pre-distribution in sensor networks which uses symmetric bi-variate polynomials to generate symmetric keys for set of sensor nodes deployed in a grid construction that provides certain connectivity. To improve such connectivity, we introduce the plat deployment method that guarantees a higher connectivity and maintains the security. We analyzed the performance of our modification on the side of the required computation, communication, and memory. Our modification provides a scheme for promising applications that require much connectivity for larger networks.

References

[1] Culler, D., Estrin, D., Srivastava, M.B.: Overview of sensor networks, pp. 41–49. IEEE Computer Society Press, Los Alamitos (2004)
[2] Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks (2002)
[3] Newsome, J., Shi, E., Song, D.X., Perrig, A.: The sybil attack in sensor networks: analysis & defenses. In: IPSN, pp. 259–268 (2004)
[4] Zhang, Q., Wang, P., Reeves, D.S., Ning, P.: Defending against sybil attacks in sensor networks. In: ICDCS Workshops, pp. 185–191 (2005)
[5] Parno, B., Perrig, A., Gligor, V.D.: Distributed detection of node replication attacks in sensor networks. In: IEEE Symposium on Security and Privacy, pp. 49–63 (2005)
[6] Pietro, R.D., Law, Y.W., Etalle, S., Hartel, P.H., Havinga, P.: State of the art in security of wireless sensor networks. IEEE Computer 35, 1–10 (2002)
[7] Perrig, A., Stankovic, J., Wagner, D.: Security in wireless sensor networks. Commun. ACM 47, 53–57 (2004)
[8] Tillet, J., Ziobro, J., Sharma, N.K.: Secure wireless sensor networks: Problems and solutions. Journal on Systemic, Cybernetics and Informatics 1, 1–11 (2004)
[9] Liu, D., Ning, P.: Establishing pairwise keys in distributed sensor networks. In: ACM CCS, pp. 52–61 (2003)
[10] Blom, R.: An optimal class of symmetric key generation systems. In: Beth, T., Cot, N., Ingemarsson, I. (eds.) EUROCRYPT 1984. LNCS, vol. 209, pp. 335–338. Springer, Heidelberg (1985)
[11] Blundo, C., Santis, A.D., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M.: Perfectly-secure key distribution for dynamic conferences. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 471–486. Springer, Heidelberg (1993)
[12] Du, W., Deng, J., Han, Y.S., Varshney, P.K., Katz, J., Khalili, A.: A pairwise key predistribution scheme for wireless sensor networks. ACM Trans. Inf. Syst. Secur. 8, 228–258 (2005)

[13] Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: ACM CCS, pp. 41–47 (2002)
[14] Chan, H., Perrig, A., Song, D.X.: Random key predistribution schemes for sensor networks. In: IEEE Symposium on Security and Privacy, p. 197 (2003)
[15] Hwang, J., Kim, Y.: Revisiting random key pre-distribution schemes for wireless sensor networks. In: SASN, pp. 43–52 (2004)
[16] Du, W., Deng, J., Han, Y.S., Chen, S., Varshney, P.K.: A key management scheme for wireless sensor networks using deployment knowledge. In: INFOCOM (2004)
[17] Ito, T., Ohta, H., Matsuda, N., Yoneda, T.: A key pre-distribution scheme for secure sensor networks using probability density function of node deployment. In: SASN, pp. 69–75 (2005)
[18] Liu, D., Ning, P., Li, R.: Establishing pairwise keys in distributed sensor networks. ACM Trans. Inf. Syst. Secur. 8, 41–77 (2005)
[19] Liu, D., Ning, P.: Establishing pairwise keys in distributed sensor networks. In: ACM CCS, pp. 52–61 (2003)
[20] Mohaisen, A., Nyang, D.: Hierarchical grid-based pairwise key pre-distribution scheme for wireless sensor nets. In: EWSN, pp. 83–98 (2006)