Hierarchical Grid-Based Pairwise Key Predistribution Scheme for Wireless Sensor Networks

Abedelaziz Mohaisen and Dae-Hun Nyang*

Information Security Research Laboratory,
Graduate School of IT & Telecommunications - Inha University,
253 YongHyun-dong, Nam-ku, Incheon 402-751, Korea
asm@seclab.inha.ac.kr, nyang@inha.ac.kr
http://seclab.inha.ac.kr

Abstract. Wireless Sensor Network (WSN) consists of huge number of sensor nodes which are small and inexpensive with very limited resources. The public key cryptography is undesirable to be used in WSN because of the limitations of the resources. A key management and predistribution techniques are required to apply the symmetric key cryptography in such a big network. Many key predistribution techniques and approaches have been proposed, but few of them considered the real WSN assumptions. In this paper, we propose a security framework that is based on a hierarchical grid for WSN considering the proper assumptions of the communication traffic and required connectivity. We apply simple keying material distribution scheme to measure the value of our framework. Finally, we provide security analysis for possible security threats in WSN.

1 Introduction

Sensors are inexpensive and low-power devices with limited resources[1]. They are small in size and have limited wireless communication capabilities with short coverage distance. The typical sensor node contains a power unit, a sensing unit, a processing unit, a storage unit and a wireless transceiver (T/R). Wireless Sensor Network (WSN) contains a huge number of sensor nodes which have limited storage and mobility. The concept of micro-sensing and wireless connection of the sensor nodes promise many new applications into military, environment, health and many other commercial areas [1]. Due to the different constraints of WSN resources, the public key cryptography algorithms such like Deffie-Hellman key agreement [6] or the RSA Signature [17] are undesirable to be used in WSN. Also using any of those will cost tens of seconds up to few minutes [5] which will expose a vulnerability to Denial of Service (DoS) attacks [19]. However, many efforts to modify the current public key cryptography to be used in WSN are still in progress.

* This work was supported by the Korea Research Grant founded by Korean Government (R03-2004-000-10023-0). D.-H. Nyang is corresponding author.

K. Römer, H. Karl, and F. Mattern (Eds.): EWSN 2006, LNCS 3868, pp. 83–98, 2006.

For the same reason of constraints, the symmetric key cryptography that uses the same key for encrypting and decrypting the messages is used in the WSN. Due to the nature of the different WSN applications, the main issue in the symmetric key cryptography is how to distribute the secret key or the keying material among different sensor nodes in WSN [8]. Since the sensor nodes are sometimes unreachable and undesirable to be updated after the deployment, many key predistribution schemes - that assign and distribute keying material or secure keys in offline phase - have been proposed. In the following section, we will review some of those schemes attached with our main contribution and with the related work.

1.1 Background Schemes and Related Works

Key Predistribution (KP) mechanisms have been the topic of active research, and many researchers have made their own appearance in the past few years. A detailed survey of such schemes is provided by Camtepe and Yener in [4]. The early scheme of the KP in WSN is Eschenauer-Gligor Scheme [8] (will be referred as **EG**). In this scheme, each node is let to randomly pick up a set of keys S_k (keys ring) out of a big pool of keys P. After the sensors are deployed, the different S_k of the different nodes provides a probabilistic connectivity value p_c, in which two nodes share a secure key (SK). When a key establishment (KE) is required for nodes i, j, the shared key $k : k \in S_{k_i} \bigcap S_{k_j}$ can be used. For those nodes that don't have a shared SK, a path key establishment (PKE) through an intermediate node is performed. To increase the resiliency of EG Scheme, Chan et al[5] proposed their upgrade on EG Scheme, where the keys pool is redesigned and the key rings S_k are drawn from the main pool with *q-composite*. When the secure KE is required, only if there are q-shared keys $k_1, \ldots, k_q \in S_{k_i} \bigcap S_{k_j}$ where S_{k_i} is the ring of the node i, then $hash(k_1||k_2......||k_q)$ is used as the secure key. If n less than q keys are shared, the two nodes perform KPE phase through an intermediate node or more.

Another scheme is Blom[2]. In this scheme, it's allowed for any pair of nodes $(i, j) \in N$ to have their own shared SK using their own keying material. The highest connectivity in a network of size N can be when using different secure keys for each outgoing path from the node itself, and a possible representation of the keys could be within a symmetric matrix of size $N \times N$. In[2], the author proposed a private D and a public G matrices to generate this matrix. A public matrix G of size $(\lambda + 1) \times N$ and a private symmetric matrix D of size $(\lambda + 1) \times (\lambda + 1)$ is defined where D entries are randomly generated. The matrix A is defined as $(DG)^T$ of size $N \times (\lambda + 1)$, and each node in the network has its corresponding row in matrix A and column in G. If the secure key is required between two nodes i, j, then the i^th row in the matrix A and the j^{th} column in G are selected and multiplied to generate one key value used as a shared SK.

In Du et al. [7] which is mainly based on EG[8] and Blom[2], the data to be stored in each node is the corresponding row and column (A_{R_j} and G_{C_j} respectively). A *multi-space* scheme was proposed considering τ number of private matrices D selected randomly out of ω pre-constructed matrices, and different

A's are created using the different D. τ rows of the different A are selected and loaded for each node. When an SK is required for two nodes(i, j), firstly *a key agreement phase* is performed on the space to be used. If there is any common space $\tau_{i,j} : \tau_{i,j} \in \tau_i, \tau_j$, then the rest of Blom scheme is continued, else, *PKE phase* using an intermediate space is performed. An intermediate node or more are used in this phase. The memory required and the computation to recover the keying material and the communication required for publishing the spaces IDs is much more than any other scheme. The connectivity provided by **EG** scheme is relatively higher than in [7]. However the resiliency in [7] is better than **EG** scheme.

Blundo et. al[3] proposed their scheme to find a method for distributing an SK in a *dynamic conference environment* using *polynomial keying material.* A Symmetric Bivariate Polynomial (SBP) of degree t is used and its shares are distributed among the parties. The polynomial uses some unique seed for each party (e.g. the communicating parties' IDs) for its variables evaluation to generate the different secure keys required to perform a secure connection in the network.

$$f(x, y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j, (a_{ij} = a_{ji}) \tag{1}$$

$$g_{ID}(x) = f(x, ID) \tag{2}$$

In a predeployment stage, each node gets the evaluated SBP $g_{ID}(x)$ in (2), and in the KE pahse, the second party node ID evaluates the second SBP variable to create a shared SK. An efficient way to implement Blundo scheme with reduced computation is in S. Schmidt et al [18].

In Liu-Ning Scheme [12], a two dimensional deployment environment constructing a grid was proposed where different nodes are deployed on different intersection points of the grid. The early discussed Blundo Scheme [3] is applied on each column and row in the grid with different SBP. Since any two nodes belonging to the same row or column have the same SBP, Blundo scheme can be directly performed on those nodes to establish a shared SK. In case there are no direct keys (i.e. $R_i \neq R_j \ and \ C_i \neq C_j$), an intermediate node is used in PKE phase. This scheme, even if it seems to be simple, provides a good connectivity providing a high probability to establish a secure direct connection. Even in the case of compromising some nodes, the network still has the ability to survive by establishing an SK using alternative nodes. On the other hand, it is possible to determine if the node can establish an SK with other nodes or not, which reduces the communication overhead. However the computational power required for this scheme is relatively high for computing $t + 1$ SBP evaluations. Using an intermediate node for key path is not efficient of computation and communication overhead.

In addition to [8][5][3][12][2][7] which we already discussed. [16], [3] proposed two security architectures. In [16], security architecture was specifically designed for the sensor networks by the name of SPINS. In SPINS, each sensor node in the network shares a secure key with a *Base Station.* Any two nodes that would like to construct a direct path could do that only using the *base station* as *Trusted Third Party* to set up a secret key. In [18], a security architecture is also proposed based on [3].

In this paper we will show the resiliency of our proposal by analyzing attacks against one node, basic zone and the whole network. The structure of this paper is: related works and notations, our scheme details, analysis and conclusion and future works.

1.2 Our Main Contribution

In this paper, we introduce a new scheme using the *Hierarchical Grid* as a deployment framework and Blundo scheme as keying material generator. Through this paper, our main contribution is to:

- Provide a scalable and robust novel framework for the key predistribution giving a perfect connectivity value to establish a pairwise key.
- Optimize the use of the different WSN resources, mainly, communication overhead, memory usage and computational power.
- Analyze and provide a theoretical and mathematical proof of our scheme performance.
- Provide and discuss the alternative against any possible security attack against our scheme.

In our scheme, we use different deployment zones which are more proper to WSN and guarantee a perfect connectivity. In addition, a higher growth in the sensor network requires much smaller keying material to be added to the original network. The network zones are constructed using a hierarchical grid that requires small fraction of information to be represented and keep tracking of the node location. Our scheme is built on Blundo[5] with some modification for generating the key material using different polynomials for the different network zones. We use different polynomials for different sections of the network to take the merit that compromising of *(*t+1) nodes will only affect the related polynomial zone, and thus, using different polynomials with different t degrees will lead to higher survivability against attacks and to exact amount of computation according to the security level.

1.3 Notations and Definitions

The following definitions and notations will be used throughout the rest of this paper

1. **Definitions**
   - ***Network Order*** n**.** Network design parameter that declares the size of the network and the number of SBP used in each sensor node.
   - ***Basic Grid or Basic Zone.*** The set of sensor nodes in a geographical area that use the same polynomial of degree t_0.
   - ***Polynomial Order*** O**.** An integer that decides the scope where the polynomial is used to establish a secure pair-wise key, $O \in \{1, 2, \ldots, n\}$. Each node has some minimum order and maximum order of n.

- ***Polynomial Degree*** t_0**.** A security parameter providing the strength of the polynomial and expressing how many distinct nodes carrying shares of this polynomial should be compromised to be able to recover the polynomial itself. The subscription 0 to n expresses the order of the polynomial.

2. **Notations**
   - n: The network order
   - N: Number of sensor nodes in the WSN
   - m: Number of sensor nodes in the Basic Network Zone
   - k: Sensor nodes distribution unit through the network
   - B_z: Basic Zone (Basic Grid)
   - O_x: Order of the x^{th} network grid
   - t_0: Degree of the basic polynomial in the basic grid
   - t_n: Degree of the polynomial for grid of order n
   - i, j: Sensor Nodes
   - ID_i: Identifier of the sensor node i
   - G_n: Number of the Basic Zones in the network

2 Hierarchical Grid-Based Scheme for Pairwise Key Predistribution

Our Scheme uses Blundo[3] as keying material generator to generate an SK for two nodes. The distribution of the keying material is performed on a *Hierarchical Grid* as in Fig. 1. Note that the hierarchical grid has been already used for a robust routing technique in the ad-hoc networks[11]. Our Modification on the grid relies on the growth factor of the network where we use the duplication as the growth factor. In addition, using different SBPs in the same sensor node to establish an SK increases the opportunity to establish a key and communicate with other nodes even if big fraction of nodes is compromised. In the following subsection, we will provide a description of our scheme.

2.1 HGB Scheme Overview

Consider a network consisting of N nodes. The different nodes are allocated as in Fig. 1, and the network is divided into n hierarchical orders of grids. Each order l consists of 2^{l-1} B_z, and the basic zone B_z is bounded by the distribution dimensions $[2k, 2k]$, where k is a uniform distribution unit of the sensor nodes in the WSN. The number of the nodes m in B_z is $(2k)^2$. The highest order O_n contains G_n=2^{n-1} basic grids. The total number of nodes is $N = m \times G_n = (2k)^2 \times 2^{n-1}$. As shown in Fig. 1, B_z is any grid with the dimensions [$G1X$,$G1Y$] which has O_1. The dimensions [$G2X$,$G2Y$] will be considered for the order 2 grids, and [$G3X$,$G3Y$] will be considered for O_3.

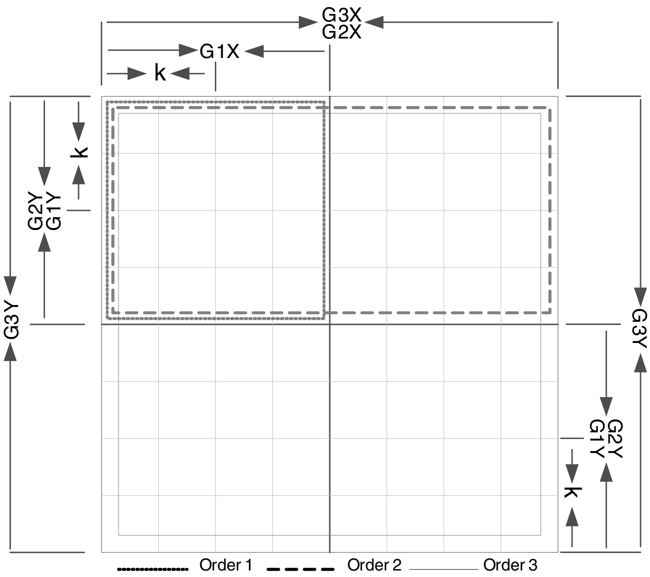

Fig. 1. Sensor nodes deployment in a hierarchical grids network

2.2 Node Identifier

Our scheme uses a smart identification material ID which is unique for each node through the network. The function of this ID is to represent the node location in the grid based network as well as the keying material required to establish an SK between two nodes just by comparing their IDs. The use of the hierarchical grid (HG) with a duplicating growth factor makes it possible to represent the different B_zs as in Fig. 2, where the leaves of the binary tree are the m nodes of the different B_z and the height of the tree is the network order. The allocated value in the end leaf is a sequence number (local ID in a B_z), where $1 \leq ID_{local} \leq m$. The different polynomials are allocated to the internal nodes of the tree. In the tree, left branches have a bit "0", and right branches have "1". The final sensor node ID is the path tracing binary string from the root node which represents the O_n polynomial down to before the leaf which represents the belonging B_z concatenated with the local ID. This structure of ID is shown in Fig. 3, and its length in bit is also shown in (3).

$$|ID| = n + \log_2 m \tag{3}$$

When the network size N is large enough, n can be considered a constant value for a robust design accepting dynamic growth of the network.

How to use this ID to establish an SK using the proper polynomial with the proper t-degree will be shown in the following subsection.

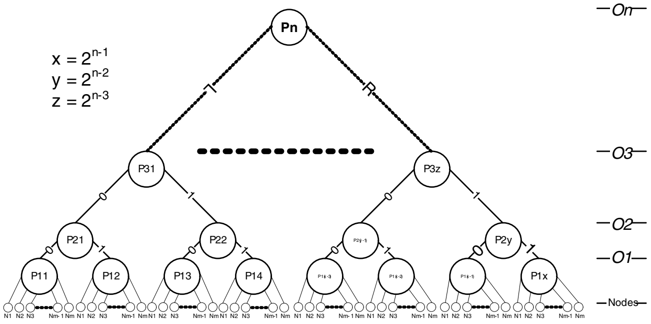

Fig. 2. Node ID Generation Determining the Node Location in the WSN

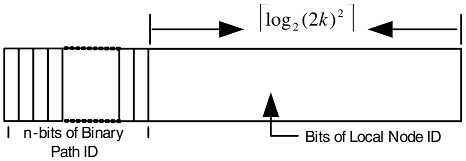

Fig. 3. Sensor Node ID Structure in the Hierarchical Grid Scheme

2.3 Key Material Generation

Using the HGB deployment structure in WSN as in Fig. 1 requires more than one keying material for each order of the network. Basically, SBP of the degree t_0 which is assigned for B_z is used to establish an SK for the pairs of nodes within the same B_z. Using this polynomial will provide a value of $\frac{1}{2n-1}$ direct key establishment opportunity out of the total possible in the network. The other polynomials are used for the connectivity to reach the perfect one. Considering that the highest amount of traffic for the communication is within B_z and other traffic fractions has a much attenuated probabilistic value, we can use polynomial of small degree for small grids, and polynomials of large degree for big order. Algorithm 1 shows the process of generating the different order of polynomials, polynomials evaluation and assignment for each sensor node.

Algorithm 1. Polynomials Generation

```
Input: Network Order n, Path ID for node i (d), Node ID
Output: 2^(n-1) Polynomials sorted in P[n][2^(n-1)],
        n SBP for each node x in Px[n] evaluated in one variable
Loop 1: for i=1 to n
      Loop2:  for j=1 to 2^(i-1)
            p[i][j]= SBP of degree t_0 and belonging to order i
            \\ Generated with coefficients belong to GF(q)
            Next j
      End Loop2
      Loop 3: For x=1 to N
            Set Px[i]=P[i][d*(1/(2^(i-1)](ID,y)
            Next x
      End Loop 3
      Next i
End Loop1
```

2.4 Secure Key Establishment

Assume that two nodes i and j would like to establish a pairwise key to communicate with each other. Each node already has its ID in addition to the other node's ID which can be exchanged before the KE phase. Firstly, a polynomial $f^*(x, y)$ is selected out of the shared polynomials in the two nodes. The selected polynomial must be common in both nodes with the minimum *t-degree* on the scope of i, j. To establish the secure key, Algorithm 2 is applied. Note that, this algorithm is applied in both i, j to generate the pairwise key. In addition, the evaluated polynomial of Algorithm 2 is in Equation (2), where x-variable is already evaluated in Algorithm 1.

Algorithm 2. Key Establishment

```
Input: IDi, IDj, Px[n]: array of node's polynomials
Output: kij, kji.
procedure:
      Begin
            Set IDi1 = Path ID of IDi
            Set IDj1 = Path ID of IDj
            Loop 1: for d = n to 1
                  if (IDi1[d] == IDj1[d])
                        Set d = d-1
                        Next d
                  Break
                  Else
                        Break
            End Loop 1
            Use f*(x,y)=Px[d]
            Set Kij=f*(IDi,IDj)
      End
```

2.5 Scheme Variables Assignment

Variables in our scheme are the degree of the polynomial t_0 and the relationship between t_0 and other polynomials degrees. The number of the nodes m in B_z is also variable. The number of the B_z itself is decided by assigning some value for the order n. In [18], the authors assigned t_0 to be 20. However, this assumption doesn't provide correlated dymanic security strength with the change of WSN size. Using the same value of memory as in [9], t_0 can be assigned as $0.6 \times m$ and the other (n-1)-polynomials' degrees $t_1, t_2, \cdots, t_{n-1}$ to follow one of the following approaches: (i) To assign the value of t_0 and the growth of the network order will lead to the same value of the polynomial growth. (ii) To consider the different t degrees independently. In our scheme, we used independent values of t for the different orders and in the analysis we calculated a dependency relations between the different t for a general use.

3 Scheme Analysis

In this part, we will focus on the performance of our scheme on two sides: The overhead analysis and the security analysis. To measure the value of the performance in our scheme, we derived mathematical formulas using the different scheme variables to express the usage of the WSN resources: memory, computation and communication. In the second hand, we follow the security analysis of Blundo[3] in terms of the compromising effect on non-compromised nodes using probablistic attacking model. We study the effect of a selective attack on our scheme compared by [8] and [12]. Node replication attack[15] and the Sybil attack[14] are mentioned. Finally, we conclude the security study by the DoM Attacks [13], DoS Attacks[19].

3.1 Overhead Analysis

Our scheme uses the different resources of the WSN in a reasonable meaning. The reduction in using any resource could affect other correlated resources and down the performance. In this section, we measure the cost of our scheme by analytical and mathematical formulas in terms of the WSN resources. From the details above, the total WSN capacity N is

$$N = 2^{(n-1)} \times (2k)^2 \tag{4}$$

Where n is the largest polynomial order in the network and k is the distribution unit of nodes, the relationship between B_z dimension and distribution unit k and the network order n for different network size is shown in Fig. 4.

Memory Overhead. The amount of memory to represent the ID in equation (3) and the different n-polynomials is required for each node. For any SBP $f(x, y)$ of degree t_0 whose coefficients in $GF(q)$, $(t_0 + 1) \times log(q)$ bits are enough to represent this polynomial. For the memory use, we have two approaches: (i) To make the degree of the polynomials independent from each other and have the same t_0 with some neglected increment in the calculations by assigning the first $t_0 = 0.6 \times m$. (ii) To make the growth of n be the same as that of t. Then, the first order takes t_0 degree and any i^{th} order could have $2^{i-1} \times t_0$. The first case cost is represented in equation (5). The first two terms are for the ID representation and the third term for n-polynomials

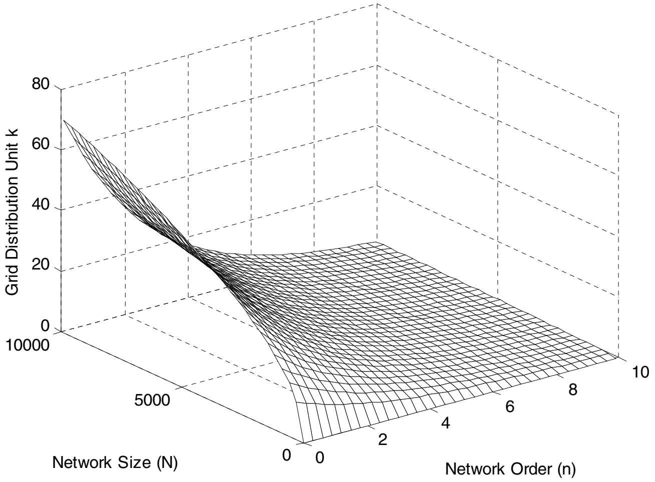

Fig. 4. The relationship between nodes distribution unit k representing the B_z size and n: the network order for different network size (N)

representation. The second case is in (6) where the third term is the summation of the required memory to represent n polynomials of different degree. The memory growth of the first case is shown in Fig. 5.(A) and the second case is in Fig. 5.(B). P_{weight} represents the bits required to represent $f(x, ID)$ of t_0 degree:

$$Memory_1 = n + \lceil log_2 \frac{N}{2^{n-1}} \rceil + n(\frac{0.6N}{2^{n-1}} + 1)log_2(q) \tag{5}$$

$$\begin{aligned} Memory_2 &= n + \lceil log_2 \frac{N}{2^{n-1}} \rceil + P_{weight} \sum_{i=1}^{n} (2^{i-1}) t_0 \\ &= n + \lceil log_2 \frac{N}{2^{n-1}} \rceil + (\frac{0.6 \times N}{2^{n-1}})(2^n - 1)(\frac{0.6N}{2^{n-1}} + 1)log_2(q) \end{aligned} \tag{6}$$

Computation Overhead. In each time an SK is required, one evaluation an SBP $f(x, ID)$ of t-degree is performed. However, the t of $f(x, ID)$ differs depending on O_x. In case of using (5), the required compuatation is one evaluation of $f(x, ID)$. In case of using (6) by assigning different ts with growth, (7) expresses the required computation in terms of the number of multiplications in $GF(q)$, where c is computation power required for two binary strings comparison representing the polynomial path identefier part, p_i is the probability that two nodes reside in different $(i-1)^{th}$ grids and CP_{t_i} is the required power for the i^{th} order SBP evaluation. Fig. 6 shows this growth curve in terms of the number of multiplications in $GF(q)$.

$$CP_{avg} = \sum_{i=1}^{n} (p_i CP_{t_i}) + c \tag{7}$$

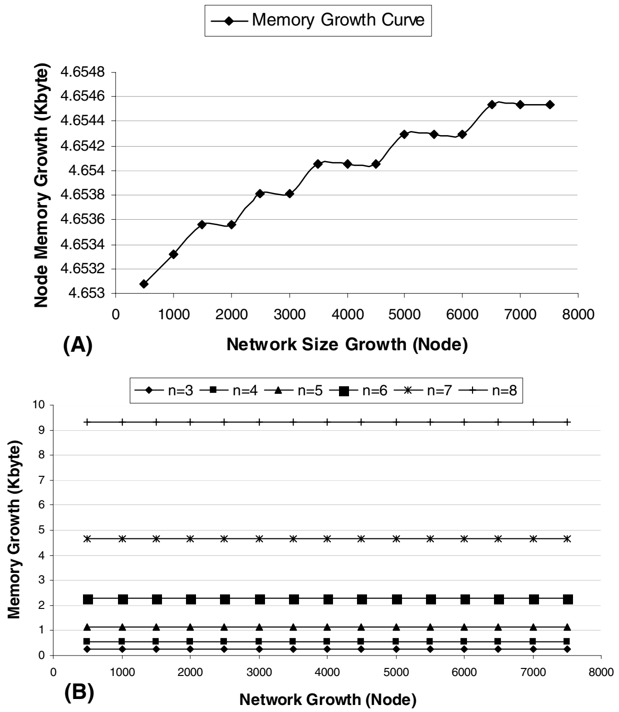

Fig. 5. (a) Network growth versus memory growth when using $n = 7$ and k to be variable. (b) Memory usage growth in the WSN when using different values of n and the same number of nodes per B_z and following the formula 2 of memory usage.

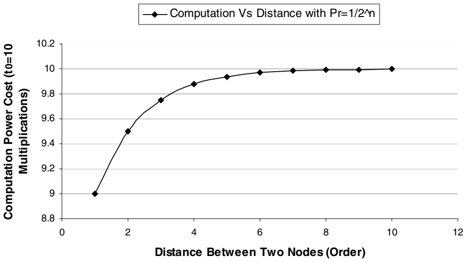

Fig. 6. Computation Overheard growth considering the communication attenuation traffic factor of $1/2^{n-1}$ versus the distance between the nodes in O

Communication Overhead. Our system does not require any extra meaning of communication. Since the different polynomials are distributed for the WSN nodes in predeployment phase, the communication overhead in the network could be in the ID exchange to construct a pairwise key. The data space required to represent the node ID is Equation (3) bits, so the required communication overhead to exchange it is bounded by the log_2N which is the space for representing N-sized network.

3.2 Security Analysis

Connectivity. Our scheme is divided hierarchically to provide a connectivity using more than one SBP for different B_z. The total provided connectivity C among the whole network is about 1. The polynomial for B_z provides connectivity of only $\frac{m}{m\times 2^{n-1}} = \frac{1}{2^{n-1}}$. Also, the polynomial for the i^{th} order grid provides connectivity of $\frac{m\times 2^{i-1}}{m\times 2^{n-1}} = \frac{1}{2^{n-i}}$. Thus, a node can establish a shared key with any node always with connectivity 1.

Blocked Communication Traffic Fraction. Basically, this paper presents a new framework for the key management in WSN. When we applied Blundo's scheme [3], we obtained that even though the i^{th} order SBP where $1 < i \leq n$ is compromised, this will not affect the other network any more than the amount of traffic (links) within the i^{th} order grid. Assume the i^{th} order SBP is compromised. Then, the fraction of the blocked traffic will be $\frac{m\times 2^{i-1}}{m\times 2^{n-1}} \times p_i = \frac{1}{2^{n-i}} \times p_i$ where p_i is the fraction of traffic between nodes resides in different the $(i-1)^{th}$ order grids. Using the current $p_i = \frac{1}{2^{i-1}}$ distribution will guarantee that the blocked communication is always constant value regardless to i value.

Compromising effects and resiliency strength. The attacking scenarios against the network can be one or more of the following:

- **An attack against N_c Nodes:** In case of compromising a set of nodes whose size is N_c that is less than t_0, the fraction of the affected nodes by those compromised ones is 0, even if all of the nodes belong to the same B_z even assuming a selective attack [9].
- **An attack against B_z:** Assume that a set of nodes s where $t_0 < |s| \leq m$ are compromised. If at least $t_0 + 1$ nodes from s belongs to one B_z, then it will lead to compromise the polynomial of the B_z. However, this seems to be so hard since the network contains 2^{n-1} polynomials and the probability p_r for t_0 nodes to be belonging to the same polynomial shares is:

$$p_r = 1 - \sum_{i=0}^{t_0-1} \binom{N_c}{i} \left(\frac{m}{N}\right)^i \left(\frac{N-m}{N}\right)^{N_c-i} \qquad (8)$$

- **An attack against the whole network:** The attack against the whole network can't be in synchronized way. However, in the worst case, it's possible to compromise the whole network by compromising all of $f(x,y)$ of t_0 one by one. To compromise B_z requires t_0 nodes to be compromised. Since the network consist of G_n different B_z, it requires to compromise $G_n \times t_0$ which is a big fraction (i. e more than 60% of the network size). Without this value of compromised nodes, the fraction of affected nodes is still less than 50% of the sensor nodes.

- **Selective Versus Random Node Attack**[9]**:** Even if the nodes are deployed in a random environment, the knowledge of the nodes deployment and the assigned polynomials for each group and the ability to distinguish the different nodes based on their B_z will lead to a selective attack. Fig. 6(B) shows the effect of this attack on WSN. In the second hand, the ranodm node attack follows probabilistic model as in (8) and differs in that m varies on intervals of $t_0, t_1 \ldots t_{(n-1)}$ to be $m_1, m_2 \ldots N$.
- **Sybil and Node Replication Attacks**[14][15]**:** There are two problems belonging to the dynamic growth of WSN. Sybil attack is done illegally by using more than one ID for the same node j. Node replication attack is performed using the same ID more than one time in the network. Our framework resists in front of those threats because it requires a structured ID that has uniform and unique structure over the entire network. When an attacker fabricates a structured ID, it should follow limited structure of our own and to be depolyed in specific area to communicate with the same B_z.
- **DoM and DoS attacks:** Denial of Messages[13] is the ability of some nodes (the attacker nodes) to deprive others of receiving some broadcast messages. Our framework doesn't require node of any broadcast capability. If any, it'll be mainly used in the same grid and thus, this attack will only affect a small fraction of the whole network. An example of the Denial of Service[19] is "attempts to prevent a particular individual from accessing a service" and this mainly happens owing to heavy communication or computation because of the keying material or any outside reason like attacker messages flooding. However, for the first case it's hard to apply DoS on our system since all the computation and communication operations are small, and take short time. In the second case, to perform a DoS, node replication attack is required.

Recovery from Compromising. When $t+1$ nodes are compromised, an alternative secure SBP will be used. In the case that an SBP of the c^{th} order grid is compromised, the SBP for the $(c+1)^{th}$ order grid is used till the system recovery and assigning another polynomial to the affected grid. Even when the highest order polynomial is compromised, the amount of traffic compromised will be only $\frac{1}{2}p_{(i=n)}$, where p_i is the fraction of the traffic between nodes that resides in different $(n-1)^{th}$ order grids. If we assume that the fraction is decreased by half whenever the order of grid increases by 1, for p_n will be $\frac{1}{2^{n-1}}$. However, the internal network connectivity will not be affected, and more than that, the majority of the secure traffic in the network will not be broken since the deployment framework guarantees that most of the traffic is in B_z.

4 Comparison with Other Schemes

We selected GBS[12], Multi-space[7], EG[8], Q-Composite and RPS [5] for the comparison with our scheme. The compared features are communication, computation and memory. The comparison for security is also shown, which is the

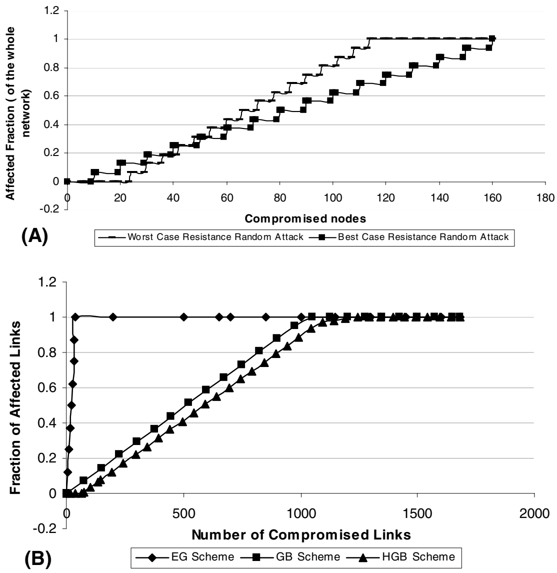

Fig. 7. (A) Worst Case Vs Best Case Selective Attack on the HGBS. (B) HGBS Selective Attack comparison with EG [8] and GBS[12] - $N = 1680$ Nodes, $t_0 = 0.6m$ for EG scheme $d = 40$.

fraction of affected non-compromised links between compromised node to the number of the compromised nodes. Table 1 shows the first part of the comparison. In our scheme, both computation and memory are shown in Equations 5, 6, and 7.

To evaluate the degree of security of our scheme and compare it with others, we used a network of 1680 nodes and mainly compared the security with GBS[12]. Parameters used in the analysis are: $N = 1680$ nodes, $n = 8$ orders, $m = 14$ nodes, $|ID| = 12$ bits that provides 7% dynamic extension of the real size ($\lceil log_2 m \rceil = 4 \mapsto m_u = 16$, dynamic extension is $\frac{(m_u - m) \times G_n}{N} \times 100\%$), $G_n = 128B_z$. For the fair comparison of GBS[12], we used the same network

Table 1. Comparison between our HGB Scheme and other Schemes: GBS of Liu Ning, EG, RPS for Chan et al. q-composite of Chan et al. and Muli-space of Du et al.

	Communication	Computation	memory
GBS[12]	Constant	SBP Evaluation	ID+2 SBP
EG[8]	$ClogS_k$	$\frac{(2C+p-p_k)}{2}logC$	S_k keys
RPS[5]	Constant	c	S_k keys
Q-Composite[5]	$ClogS_k$	$ClogC$ Comparison	S_k keys
Multi-space[7]	$Clog(n \times \tau)$	2 Vectors Multiplication of size λ	$\tau + 1$ of size λ Vectors
HGBS*	Constant	SBP Evaluation	ID+n SBP

size and $m = 41$. We applied the test of the fraction of affect non-compromised nodes between compromised nodes using a selective attack. Fig. 7 (A) shows this comparison. Fig. 7 (B) shows two cases of attacking against our scheme: *Selective attacking* with the worst case consideration and *Selective attacking* with the best case consideration. The whole compromising growth in our scheme and the GBS is the same, but our scheme is right-shifted by $n \times t_0$.

Remark: The constant value of communication in GBS depends on whether it's possible to construct a direct key or not. In case of using an intermediate node, the communication cost of the intermediate should be considered. The amount of communication traffic in our scheme is always constant because of its nature.

5 Conclusion and Further Work

In this paper, we proposed a novel proper framework for the secure key management and predistribution in the WSN. We proposed hierarchical grid for the sensor nodes deployment that bounds the heavily communicated nodes in one basic grid that has strong secure keying material.

We also designed an ID structure which is unique for the node and expresses the location as well as the keying material to be used. To measure the performance of our framework, we used Blundo[3] as a keying material generator block. Mathematical analysis of the computation, communication and memory was provided. The different possible attacks were lightly touched. The performance shown comparison expressed the value of our framework.

The next works will be an enhancement for the keying material assignment on the framework, in addition to deeper security analysis and more rigorous mathematical model derivation on security. We will provide a detailed key establishment algorithm that considers the framework connectivity, communication traffic, security and design parameters.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: *Wireless Sensor Networks: A Survey*, Computer Networks (Elsevier) Journal, Vol. 38, No. 4, pp. 393-422, March 2002.
2. Blom, R.: *An optimal class of symmetric key generation systems, Advances in Cryptography*, Proceedings EUROCRYPT 84 , LNCS , Springer-Verlag, 209, pp. 335-338, 1985.
3. Blundo, C., DE Santis, A., Herzberg, A., Kutten, S., Vaccaro, U., and Yung, M.: *Perfectly secure key distribution for dynamic conferences*, In Advances in Cryptology - CRYPTO '92, LNCS 740, pp. 471-486, 1993.
4. Camtepe, S. A. Yener, B.: *Key Distribution Mechanisms for Wireless Sensor Networks: a Survey*, Rensselaer Polytechnic Institute RPI, Technical Report TR-05-07, (March 23, 2005).
5. Chan, H., Perrig, A., Song, D.: *Random key predistribution schemes for sensor networks*, IEEE Symposium on Security and Privacy, pp. 197-213, May 2003.

6. Diffie, W., Hellman, M. E.: *New directions in cryptography*, IEEE Trans. Inform. Theory, IT-22, pp. 644-654, November 1976.
7. Du, W., Deng, J., Han, Y. S., and Varshney, P.: *A pairwise key pre-distribution scheme for wireless sensor networks*, In Proceedings of 10th ACM Conf. on Computer and Communications Security (CCS'03), pp. 42-51, 2003.
8. Eschenauer, L., Gligor, V. D.: *A key management scheme for distributed sensor networks*, In Proceeding of the 9^{th} ACM Conf. on Computer and Communications Security, pp. 41-47, 2002
9. Huang, D. , Mehta, M., Mehdi, D, Harm, L.: *Location-aware Key Management Scheme for Wireless Sensor Networks*, Proc. of 2004 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04), pp. 29-42, October 2004.
10. Hwang, J. M., Kim, Y. D.: *Revisiting random key pre-distribution schemes for wireless sensor networks*, Workshop on Security of ad hoc and Sensor Networks archive, Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks, pp. 43 - 52, 2004.
11. Li, J., Janotti, J., DeCouto, D. S. J. , Karger, D. R., Morris, R.: *A Scalable Location Service for Geographic Ad Hoc Routing*, The Sixth Annual International Conf. on Mobile Computing and Networking, pp. 120-130, August 2000.
12. Liu, D., Ning, P.: *Establishing Pairwise keys in distributed sensor networks*, In Proceedings of 10th ACM Conf. on Computer and Communications Security (CCS'03), pp. 52-61, 2003.
13. McCune, J., Shi, E., Perrig, A., Reiter, M.: *Detection of Denial-of-Message Attacks on Sensor Network Broadcasts*, In Proceedings of the IEEE Symposium on Security and Privacy, May 2005.
14. Newsome, J., Shi, E., Song D., Perrig A.: *The Sybil Attack in Sensor Networks: Analysis and Defense.*, In Proceedings of Information Processing in Sensor Networks (IPSN), April 2004.
15. Parno, B., Perrig, A., and Gligor V.: *Distributed Detection of Node Replication Attacks in Sensor Networks*, Proceedings of the 2005 IEEE Symposium on Security and Privacy, May 2005.
16. Perrig, A., Szewczyk, R., Wen, V., Culler, D. E., Tygar, J. D.: *SPINS: security protocols for sensor networks*, MOBICOM, pp. 189-199, 2001.
17. Rivest, R. L., Shamir, A., Adleman, L. M.: *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM, 21(2): pp. 120-126, 1978.
18. Schmidt, JS. , Krahn, H., Fischer, S., Watjen, D.: *A Security Architecture for Mobile Wireless Sensor Networks*, Security in Ad-hoc and Sensor Networks, LNCS 3313, pp 166-177, Springer-Verlag Berlin Heidelberg 2005
19. Wood, A.,Stankovic, J. A.: *Denial of Service in Sensor Networks*, IEEE Computer, 35(10): pp. 54-62, October 2002.