Two-level Key Pool Design-based Random Key Pre-distribution in Wireless Sensor Networks

Abedelaziz Mohaisen*, *Member*, DaeHun Nyang, and Tamer AbuHmed****
*****Electronics and Telecommunication Research Institute
Daejeon 305-700, Korea
[e-mail: a.mohaisen@etri.re.kr]
******Information Security Research Lab, Inha University
Incheon 402-751, Korea
[e-mail: nyang@inha.ac.kr, tamer@seclab.inha.ac.kr]
Corresponding author: Abedelaziz Mohaisen

Received August 9, 2008; revised September 15, 2008; accepted October 3, 2008; published October 25, 2008

Abstract

In this paper, the random key pre-distribution scheme introduced in ACM CCS'02 by Eschenauer and Gligor is reexamined, and a generalized form of key establishment is introduced. As the communication overhead is one of the most critical constraints of any successful protocol design, we introduce an alternative scheme in which the connectivity is maintained at the same level as in the original work, while the communication overhead is reduced by about 40% of the original overhead, for various carefully chosen parameters. The main modification relies on the use of a two-level key pool design and two round assignment/key establishment phases. Further analysis demonstrates the efficiency of our modification.

Keywords: Sensor network, security, random key distribution, connectivity, communication efficiency

This research was supported by the MKE (Ministry of Knowledge Economy) of the Korean government under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement) (IITA-2008-C1090-0801-0028) and IT R&D program of MKE/IITA. [2005-Y-001-04, Development of Next Generation Security Technology]

DOI: 10.3837/tiis.2008.05.001

1. Introduction

Due to the promising variety of applications supported, wireless sensor networks (WSNs) are among the most active areas of research. Several challenging issues have been studied [1]. Because they are deployed in an open environment that enables vulnerabilities to many attacks [1][2], and because of resource-constraints, the security of WSNs is one of the most challenging issues [3]. To enable secure WSNs, security research has aimed to provide efficient encryption algorithms that consider network constraints [3]. For instance, several studies have considered the efficiency of both public and symmetric key algorithms in typical sensor networks [4][5][6][7]. For public key algorithms, it has recently been determined that they are efficient in a typical sensor platform to some extent, though careful utilization is recommended. In practice, public key algorithms are expected to be used for securing symmetric key algorithms distribution in sensor networks, rather than encryption and decryption of all communication traffic in sensor networks [8][9].

On the other hand, symmetric key algorithms that use the same secret key for both encryption and decryption at both communication sides, have been shown to be computationally light, fast, robust and more efficient for WSN [10][11][12][13][14][15][16]. However, due to the lack of infrastructure, the key pre-distribution (KPD) problem is considered one of the most challenging issues. In KPD, sets of keys [10][11] or keying material [13][14][15][16] are assigned to each sensor node, to enable secure communication. Most current KPD schemes have to some extent succeeded in achieving a marginal efficiency gain for securing WSN [1].

In terms of their requirements, most recently introduced KPD schemes have considered the memory constraints as a strong bottleneck in their design, and tried to minimize the memory utilization. However, there are several currently used sensor platforms that support additional memory, resulting in more flexible memory constraints [17]. One of these works is based on the random key assignment (EG scheme) in [10], which has been extended in [11] and reexamined in [18]. The importance of this work is that in addition to the memory efficiency, negligible computation is required for key generation or establishment, unlike other works based on keying material assignment [13][14]. However, one of the critical shortcomings in this work, in addition to the low resiliency, is the communication overhead required for exchanging identifiers of pre-loaded keys. In this work, we introduce a scheme based on the EG work, to overcome the problem of the communication overhead, while achieving the same benefits as the EG scheme, including local connectivity, and a reduction in the resource requirements. In the following, we introduce the notation, contributions and detailed structure of this paper.

1.1 Paper Contributions and Structure

In this article, we reexamine the random key pre-distribution (RKPD) scheme in WSN [10] and introduce a generalized scheme that has several advantages. On the one hand, the modified scheme aims to reduce the resource requirements of RKPD in terms of communication and memory. On the other hand, the modified scheme achieves the same connectivity as that required by RKPD. We provide a rigid analysis, to explore the performance of our modified scheme and compare it to RKPD, in order to demonstrate the achieved goals.

In terms of the paper structure, Section 2 introduces an overview on previous works. Section 3 introduces our protocol followed by further analysis and a demonstration of our

contributions in Section 4. Discussions of common network architectures and special cases of the scheme are introduced in Section 5. Finally, the conclusion is provided in Section 6.

1.2 Notation

The notation used hereafter is shown in **Table 1**.

Table 1. Notation

Symbol	Definition
N	Overall number of nodes in the network
I	Improved communication overhead as a percentage of the initial overhead
M	Memory overhead
c	Number of keys selected from each sub-pool for a node.
P_B, P_s	Overall number of keys in the keys' pool and number of keys in each sub-pool
S_p, S_n	Overall number of sub-pools and number of sub-pools for a node, respectively.
C_{oh}	Communication overhead due to keys' information exchange.
p_g, p_l	Global and local connectivity ($p_g = p_k p_p$)
ID_{key}	Identifier of key, where the size of the identifier is defined as $\mid ID_{key} \mid = \log_2 P_B$
ID_{S_p}	Identifier of the sub-pool with length $\mid ID_{S_p} \mid = \log_2 (P_B / P_s)$

2. Previous Works on Key Pre-distribution in WSN

Several works have been introduced to solve the problem of key pre-distribution (KPD) in wireless sensor networks. These works were not limited to random key assignment, but also included several works based on the bivariate symmetric polynomial [19] such as works in [13] [20], and symmetric matrices [21] such as works in [14][22]. In this section, we review a set of selected works on KPD, followed by a detailed describtion of RKPD. For an extensive survey on KPD schemes, refer to [23].

2.1 Selected Related Works

2.1.1 Blom [21]: Basically, this scheme was not designed for sensor networks, but has been recently used as the basis of many KPD schemes. Blom's scheme utilizes the symmetry property of the matrix in order to generate symmetric keys for different communicating parties. For instance, a symmetric matrix G is used, in which each node has its own column and row, and the corresponding columns or rows are assigned to different nodes. If two nodes s_i and s_j want to communicate securely, they use the elements E_{ij} in G for the s_i side and E_{ji} in G for the s_j side as keys. Note that $E_{ij} = E_{ji}$, since the matrix G is symmetric.

2.1.2 Du et al [14][22]: Based on Blom's scheme [21], this scheme aims to reduce its memory requirements while reducing its connectivity, by utlizing the concept of multiple space, as in the EG scheme. In both works, a public matrix G of size $(\lambda+1)\times N$, where λ is a security parameter, and a private symmetric matrix D of size $(\lambda+1)\times(\lambda+1)$, where D has random elements, are constructed. Also, $A = (DG)^T$ of size $N \times (\lambda+1)$ is defined. For a

node s_i, the row R_i in A and column C_i in G are selected. The two nodes then can communicate by exchanging their public columns and computing $k_{ij} = R_i C_j$ at the s_i side and $k_{ji} = R_j C_i$ at the s_j side. Note that both keys are equal, because the matrix A is symmetric. In the later work, an additional extension was made by utilizing the deployed knowledge [22]. For greater accuracy, different deployment structures with practical error measurements, were used by Ito et al. in [24].

2.1.3 Blundo et al. [19]: This work was not designed for WSN, but has been extensively used in the context of WSN and is based on a symmetric bivariate polynomial in the form of $f(x,y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j$, where $a_{ij} = a_{ji}$, to distribute keys for the nodes in the network. The polynomial share $g_i(y) = f(i,y)$ is calculated and stored in node s_i of ID i. For two nodes s_i and s_j, the two keys are computed as $k_{ij} = g_i(j)$ and $k_{ji} = g_j(i)$, respectively.

2.1.4. Liu et al. [13][20]: These works utilize Blundo's scheme as a foundation [19] and use symmetric polynomials for generating a symmetric key for the different nodes in the network. In [13] and [20], the EG scheme [10] is applied in a pool of polynomials, rather than cryptographic keys. In [13], a grid-based, key pre-distribution scheme that provides high resiliency and connectivity features based on [19] is also introduced.

2.1.5 Mohaisen et al. [15][16]: These works are based on the symmetric bivariate polynomial of Blundo et al. and utilize a hierarchal, grid-based deployment knowledge scenario [15] with smart nodes identification for reducing resource requirements per node on average, and using different polynomials of different size, based on their locations. Also, by modeling the communication pattern and extending the grid-based scheme in [13], they introduced a plat-based key pre-distribution and establishment scheme [16] that enabled an improved connectivity from the previous work.

2.2 Overview of the Basic Random Key Establishment Scheme

The early KPD scheme adapted for WSN was introduced by Eschenauer-Gligor [10]; it is referred to as EG for breivty. In the EG scheme, each node randomly selects a key ring S_k of size K from a large key pool of size P_B. This selection process provides probabilistic connectivity computing, by the equation:

$$p_g = 1 - \frac{((P_B - K)!)^2}{(P_B - 2K)! P_B!} \tag{1}$$

Via Stirling's approximation for $n! \approx \sqrt{2\pi} n^{n+\frac{1}{2}} e^{-n}$ [10], the aforementioned equation for connectivity can be expressed as follows:

$$P_g = 1 - \frac{(1 - K/P_B)^{2(P_B - K + \frac{1}{2})}}{(1 - 2K/P_B)^{2(P_B - K + 1)}} \tag{2}$$

Accordingly, traffic between two nodes s_i and s_j is secured via a key k shared between the two nodes as an encryption key (in other words, $S_{k_i} \bigcap S_{k_j} \neq \phi$). Otherwise, there is a path

discovery phase via one or more intermediate nodes. In [10], the memory utilization is lower than the naïve scheme that assigns the whole key pool to each node, however, the result is a weak resiliency. In other works, if a small fraction of nodes are compromised, a large fraction of communication between non-compromised nodes immediately becomes insecure, due to the exposure of their keys to an adversary. To overcome this shortcoming and improve the overall performance of the scheme, Chan et. al. proposed the Q-COMPOSITE scheme [11]. Via the same EG procedure, a key between two nodes s_i and s_j is established if and only if $S_{k_i} \bigcap S_{k_j}$ is a set of q number of keys (where q >1). If $\{k_1,...,k_q\} \in \{S_{k_i} \bigcap S_{k_j}\}$, the value $h(k_1 \| k_2,...,\| k_q)$ is used as the key at both nodes' sides. Otherwise, one or more intermediate nodes are used. Further analytical analysis and extension of the probabilistic scheme is provided in Hwang and Kim. [18].

3. Our Protocol

Our protocol consists of two main phases, viz., the offline and online phase. In the offline phase, which is performed at a pre-deployment time by an administrator, sets of keys are assigned to several nodes in the network. In the online phase, which is performed when secure communication between two nodes is needed, the two nodes determine a common shared key or establish a key path via one or more intermediate nodes. The details of the two phases are provided below.

3.1 Offline Phase: Key Generation and Assignment

The offline phase of our scheme is performed as follows:

1. **Initialization:** Keys are generated and grouped into different sub-pools according to the following steps:
   A. The administrator randomly generates P_B number of keys. Each key has a pre-determined length that provides a reasonable security level and is chosen according to the encryption algorithm used. Typical keys may have a length of 64, 128, or 256 bits.
   B. The admininstrator randomly groups generated keys into S_B number of sub-pools, where each sub-pool has P_s number of keys (where $P_s = P_B / S_P$). Each sub-pool has a unique identifier within the network, denoted as ID_{S_p} and each key within the same sub-pool has a unique identifier, denoted as ID_{Key}. The length of the key's identifier is determined by the number of keys in a single sub-pool, and such an identifier is typically re-used in other sub-pools, in order to identify other keys. Note that a single key is identified within a specific sub-pool by its own identifier ID_{Key}, and is identified globally by its identifier and the identifier of the sub-pool to which it belongs (ID_{Key}, ID_{S_p}).
2. **Key Assignment**: Keys are assigned to different nodes in the network according to the following steps:

A. The admininstrator chooses S_n number of sub-pools for each node in the network.
B. From each sub-pool assigned in the previous step for a specific node, the administrator chooses and groups c number of keys.
C. The set of keys associated with their identifiers and sub-pool identifiers are stored in the specified node's memory. For each group of c number of keys, only one sub-pool identifier is used and stored in the sensor node.

After this phase, nodes are ready to be deployed and securely communicate via the set of keys stored in their memory from the online key establishment phase.

3.2 Online Phase: Key Establishment Phase

In this phase, two sensor nodes that want to communicate securely can exchange the identifiers of their keying materials and if there is any common key, they can communicate securely via that key. The procedure for this phase includes the following steps:

1. The two nodes s_i and s_j exchange the identifiers of sub-pools from which they have keys. This explicitly requires $S_n \log_2 S_p = S_n \log_2 (P_B / P_s)$ bits of communication overhead.
2. If none of the sub-pools from which the two nodes have keys are shared, the protocol is terminated and there is a path key establishment phase, to find one or more intermediate nodes through which a key can be established. If one or more sub-pools are shared, the following procedure is performed for each common sub-pool, unless termination has already occurred by finding a common key in a previous step:
   A. The two nodes exchange identifiers of keys within the currently shared sub-pool. This explicitly requires $c \log_2 P_s$ bits of communication overhead per node, for each communication round. The number of rounds is determined by the number of shared sub-pools between two nodes.
   B. If a key is shared between the two nodes via exchanged identifiers of keys within shared sub-pools in the previous step, each node uses it as a secret key and terminates the whole process. Otherwise, the process is performed for the whole set of shared sub-pools.
   C. If after exchanging the identifiers of all keys stored in all nodes no shared key is found, a path key establishement phase is performed according to the procedure suggested in the original EG scheme.

4. Analysis and Evaluation

In this section, we provide a rigid analysis of our scheme compared to the reexamined EG scheme in [10]. We basically analyze the effect of our modification of the EG scheme on the connectivity, memory requirements, computation and communication overheads. The advantage of our scheme is obvious in two cases:

1. When two nodes do not share any sub-pool, the exchanged traffic due to key establishment is reduced to the number of sub-pools per node multiplied by the size of each sub-pool identifier. By contrast, the EG scheme requires exchanging the whole

set of identifiers of keys, after which termination ocurrs, when it is confirmed that no key is shared.

2. When two nodes share several sub-pools and the first shared sub-pool has a common key.

On the other hand, the worst-case performance of our scheme occurs when all sub-pools in two nodes are shared, but none of the keys within these sub-pools are shared among the two nodes. In this case, the communication requirements in our scheme are equal to those of the EG scheme..

4.1 Connectivity Estimation

The connectivity is classified into two types, viz., *local connectivity* and *global connectivity*. In the following, we compare our scheme with the EG scheme in both respects.

Definition (Local Connectivity) *The local connectivity (p_l) is defined as the fraction of nodes from the overall network that a given node can communicate with directly in a single-hop manner via its own keying material.*

Since the offline phase of our scheme is divided into two stages, where a random selection is performed in each stage, the local connectivity p_l is defined as the combination of both selection probabilities for a sub-pool and, thereafter, for a key to be shared between two nodes. The first component (p_p) is due to the probability that one or more sub-pools are shared between two nodes, which is realized according to the following equation:

$$p_p = \prod_{i=0}^{t-1} \frac{1}{S_p - i} \prod_{j=t}^{S_n} \frac{S_p - S_n - j}{S_p - j - 1} \text{ , for t=1 to } S_n \tag{3}$$

Assuming that there is at least one shared sub-pool between two nodes, the second probability component (p_k) is due to shared keys in the shared sub-pool, which is realized according to the following equation:

$$p_k = \frac{c}{P_s} = c \frac{S_p}{P_B} \tag{4}$$

The overall local connectivity is determined as the product of defined probabilities (i.e., $p_l = p_p p_k$), which yields:

$$p_l = \left(c \frac{S_p}{P_B} \right)^t \prod_{i=0}^{t-1} \frac{1}{S_p - i} \prod_{j=t}^{S_n} \frac{S_p - S_n - j}{S_p - j - 1} \text{, for t=1 to } S_n \tag{5}$$

Note that p_l is computed by assuming an independent selection of keys associated with links between nodes. This restricted selection process guarantees that any key is only used for a single link. However, in case we want to use keys without restrictions, Eq. 5 can be expressed as:

$$p_l = \frac{S_n}{S_p} \left(\frac{c}{P_S} \right) \tag{6}$$

Generally, if the numbers of keys selected from each sub-pool are different, where there are c_i number of keys selected from the *i*-th sub-pool of a specific node, Eq. 6 can be expressed as follows, adopting different c_i values in the final probability:

$$p_l = \frac{S_n}{S_p}\left(\sum_{i=0}^{S_n}\frac{c_i}{S_n P_S}\right) = \frac{1}{S_n P_S}\left(\sum_{i=0}^{S_n} c_i\right) = \frac{1}{P_B}\left(\sum_{i=0}^{S_n} c_i\right) \quad (7)$$

Note that p_l in the EG scheme is defined according to the number of keys assigned to each node, which are obtained by: $p_l = \frac{K}{P_B}$, where $K = \sum_{t=0}^{S_n} c_i$.

Lemma 4.1. *When the overall number of keys in our scheme is sat equal to the number of keys assigned to each node in the EG scheme, our proposed scheme achieves the same local connectivity* p_l *as that in the EG scheme.*

Proof.

Let $K = \sum_{i=0}^{S_n} c_i$, then Eq. 7 can be expressed as $p_l = \frac{K}{P_B}$, which is equal to the connectivity provided in EG scheme [10].

However, based on further memory analysis and the aforementioned identifiers' representation of the keys and their sub-pools, the memory required for keys' identifiers in our scheme is less than that in the EG scheme.

Finally, **Fig. 1** shows the local connectivity p_l for different network sizes, represented in terms of P_B versus K values. Note that p_l is proportional to the number of keys assigned to each node, K .

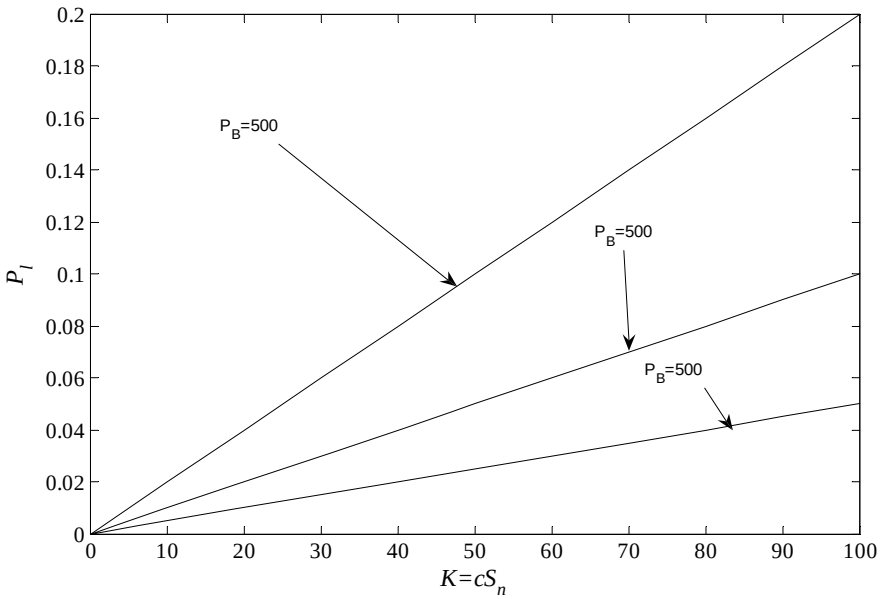

Fig. 1. Local connectivity(p_l), which is the same in both works, is proportional to the K value

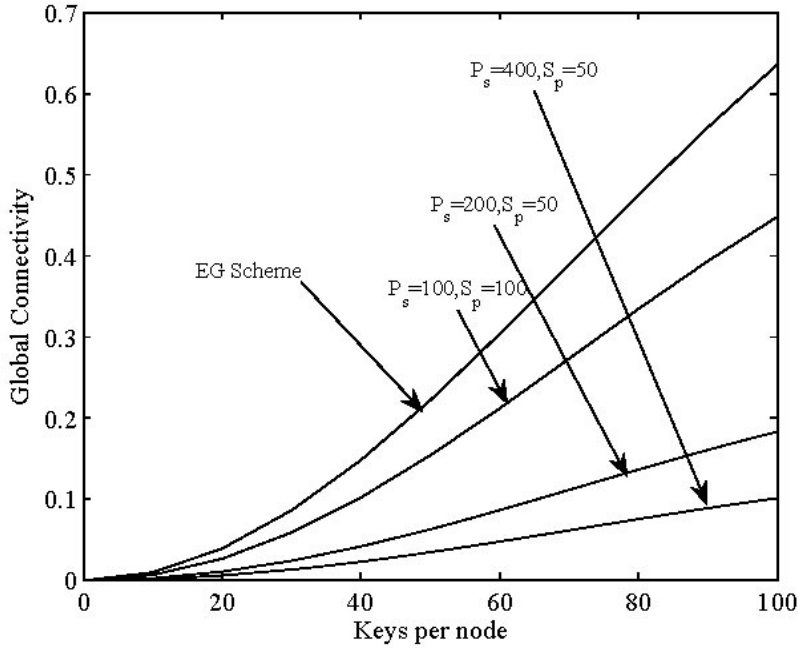

Fig. 2. Global connectivity (p_g) comparison between our scheme and the EG scheme for P_B =10,000, various K (i.e., $K = cS_n$) and various parameters

Definition 2 (Global Connectivity) *The global connectivity (p_g) is defined as the degree of connectivity of the graph $G(e,v)$, regardless of the number of hops, given the number of keys assigned to each node.*

As a result, global connectivity considers the probability of a node communicating with an arbitrary node in the network, regardless of the number of intermediate nodes used. This probability is calulated by adopting the notation given earlier and the connectivity analysis and estimation provided in Section 2.2, and is given by the following equation:

$$p_p = 1 - \frac{\left(\left(S_p - S_n\right)!\right)^2}{\left(S_p - 2S_n\right)!S_p!} = 1 - \frac{\left(1 - \frac{S_n}{S_p}\right)^{2\left(S_p - S_n + \frac{1}{2}\right)}}{\left(1 - \frac{2S_n}{S_p}\right)^{2\left(S_p - 2S_n + 1\right)}} \tag{8}$$

Similarly, we can compute p_k as:

$$p_k = 1 - \frac{\left((P_S - c)!\right)^2}{(P_S - c)!P_S!} = 1 - \frac{\left(1 - \frac{c}{P_S}\right)^{2\left(P_S - c + \frac{1}{2}\right)}}{\left(1 - \frac{2c}{P_S}\right)^{(P_S - 2c + 1)}} \quad (9)$$

Where p_p is the connectivity determined by S_n, the number of sub-pools and p_k is determined by the internal set of keys from each sub-pool. The resulting overall connectivity is given by the following:

$$p_g = p_k p_p \quad (10)$$

4.2 Overhead Evaluation

In this section, we analyze the resources required in our scheme.

4.2.1 Memory Overhead: If the number of keys in our scheme is the same as in the EG scheme, the memory required for representing these keys is also the same. The memory required for storing the different keys is determined by: $M = c \times S_n \times |K_l|$, where $|K_l|$ is the key length. In the EG scheme, the required memory overhead is $K \times |K_l|$, where K is the number of keys assigned to each node. Note that both values are equal, since $K = cS_n$.

Besides, additional memory is required for storing the identifiers of the different keys. This information is not related to the size of the key itself, but rather the number of keys assigned to each node, the overall number of sub-pools, and the number of keys within each sub-pool. Our scheme directly reduces the memory requirement for storing the identifiers, if the various parameters are carefully assigned. For instance, assuming that the number of keys selected from each sub-pool for each node are equal, the memory requirement in our scheme is computed as: $M = cS_n \left(\log_2 \frac{P_B}{S_p} \right) + S_n \left(\log_2 \frac{P_B}{P_S} \right)$. On the other hand, the EG scheme requires $K \log_2 P_B$ bits for storing the same number of keys' identifiers. This memory requirement, however, is equivalent to the worst-case communication overhead for our scheme.

To illustrate the additional memory requirement for the identifiers, **Fig. 3** shows the representation of the keys stored in each node in our scheme and the EG scheme. Note that S_n number of blocks are required for our scheme (where only one is shown). The following numerical example clarifies the gain in the aforementioned scenario.

Example 1. Consider the following parameters: N=10,000, P_B =10,000, S_p =100, K =256, $c = S_n$ =16, and $S_p = P_S$ =100. The additional memory requirement for the keys' identifiers in the EG scheme is: $(K)(\log_2 P_B)$ =(256) $(\log_2 10000)$ =3,584 bits[1]. For the same case,

[1] We use the ceiling value for $\log_2 10{,}000$.

however, our scheme requires: $(16)(16)\left(\log_2 \frac{10000}{100}\right)+(16)\left(\log_2 \frac{10000}{100}\right)$ = 1,904 bits for storing the same number of keys' identifiers.

Selecting an equal number of keys from each sub-pool and assigning them to each node enables an efficient method of discriminating the beginning and ending of the keys' blocks. In other words, no additional bits are required for bit stuffing in order to distinguish sub-pools' identifiers, keys' identifiers, and keys themselves, since all lengths can be stored in advance as a system parameters. Furthermore, schemes for efficient storage that have been introduced in [9] can easily be used to reduce the overhead in our scheme.

4.2.2 Communication Overhead: The online phase of our protocol consists of two stages. For the first stage, identifiers of S_n sub-pools, each of length $\log_2 S_p$ bits, are exchanged, requiring C_{oh}^1 bits transmission overhead, defined as:

$$C_{oh}^1 = S_n \log_2 S_p = S_n \left(\log_2 \frac{P_B}{P_S} \right) \tag{11}$$

In the second stage, internal keys' identifiers of some sub-pools are transferred, if there are some common sub-pools shared between the two nodes. The worst-case requirements for such overheads occur at S_n times of transmission requiring C_{oh}^2 bits of transmission overhead which is the sum of:

$$C_{oh}^2 = cS_nP_S = cS_n \left(\log_2 \frac{P_B}{S_p} \right) \tag{12}$$

The overall overhead is the sum of the two overhead, which yields the following:

$$C_{oh} = cS_n \left(\log_2 \frac{P_B}{S_p} \right) + S_n \left(\log_2 \frac{P_B}{P_S} \right) \tag{13}$$

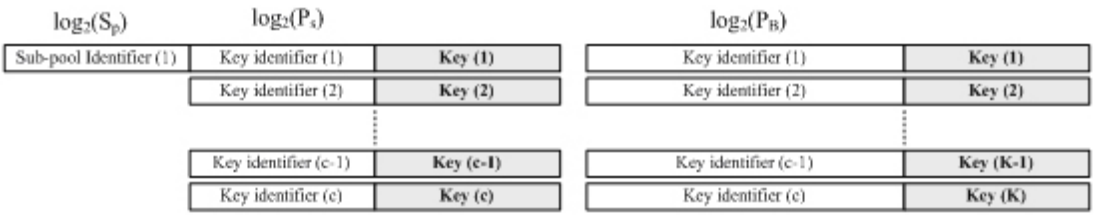

Fig. 3. Memory required for storing keys and keys' identifiers in our scheme (left) and EG scheme (right)

Lema 4.2 *For the same memory overhead, our protocol requires less communication overhead* ***on average*** *than that in the EG scheme in [10] for any c greater than a threshold of:* $\frac{\log_2 P_B + \log_2 P_S}{\log_2 S_p}$.

Proof.

Given that the overhead in the EG scheme is estimated as $C_{eg} = K \log_2 P_B$ and that in our scheme is represented by Eq. 13, we firstly assume that $C_{ours} \le C_{eg}$ and determine what parameters satisfy this inequality, given that: $K = cS_n, P_S = \frac{P_B}{S_P}$ and $\log_2 \frac{P_B}{S_p} = \log_2 P_B - \log_2 S_p$.

$$cS_n\left(\log_2 \frac{P_B}{S_p}\right) + S_n\left(\log_2 \frac{P_B}{S_p}\right) \le K \log_2 P_B$$

$$K\left(\log_2 \frac{P_B}{S_p}\right) + \frac{K}{c}\left(\log_2 \frac{P_B}{P_S}\right) \le K \log_2 P_B$$

$$\log_2 \frac{P_B}{S_p} + \frac{1}{c}\left(\log_2 \frac{P_B}{P_S}\right) \le \log_2 P_B$$

$$-\log_2 S_p + \frac{1}{c}\left(\log_2 P_B + \log_2 P_S\right) \le 0$$

$$\frac{\log_2 S_p}{\log_2 P_B + \log_2 P_S} \ge \frac{1}{c}$$

$$\frac{\log_2 P_B + \log_2 P_S}{\log_2 S_p} \le c \tag{14}$$

By assigning a c greater than the value determined above, we can ensure that the communication overhead in our scheme is always less than that in the EG scheme.

By substituting $S_P = \frac{P_B}{P_S}$ in Eq. 14, we ascertain that c needs to satisfy the following:

$$c \ge 2\frac{\log_2 P_B}{\log_2 S_p} - 1 \tag{15}$$

Alternatively, Eq. 15 implies that the EG scheme requires less communication overhead if $K \le S_n \frac{\log_2 P_B + \log_2 P_S}{\log_2 S_P}$. However, to achieve sufficient local connectivity, K must be much greater than this value, which contradicts the basic assumption. In our scheme, if we set S_p equal to $\sqrt{P_B}$, we can easily ensure that the above condition is satisfied (which ensures that c=3 at least). **Fig. 6** shows the relationship between the desired c and other parameters. **Fig. 4** shows the same case for various values of P_B.

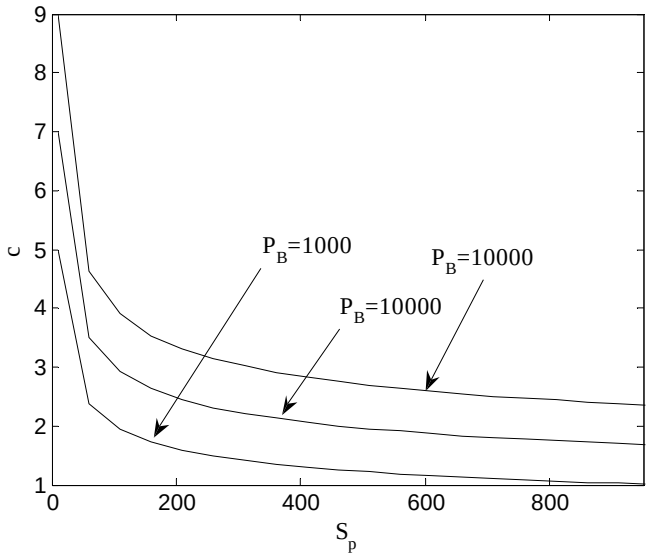

Fig. 4. c value for constant P_B in order to ensure that our scheme outperforms the EG scheme

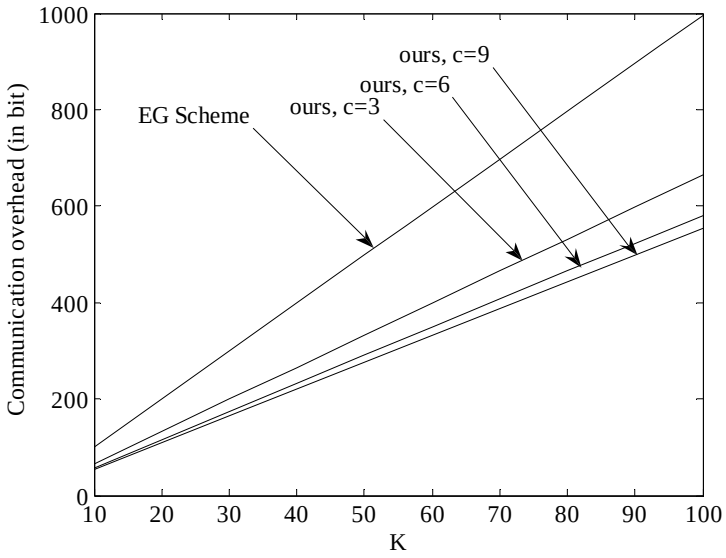

Fig. 5. Comparison of the communication overhead between our scheme and the EG scheme given that $P_S = S_p = \sqrt{P_B}$. Note that our scheme outperforms the EG scheme in terms of its communication requirements when c>2

The reduction in communication overhead as a percentage is computed as: $I = \frac{C_{eg} - C_{ours}}{C_{eg}} \times 100\%$. For example, $I = 33.33\%$ when $c = 3$, 41.67%, $c = 6$, and I= 44.44% when $c = 9$.

Fig. 5 shows a comparison between our scheme and the EG scheme, in terms of the required communication overhead in bits for various c values. **Fig. 6** shows an extended illustration of the relationship between c and S_p for various network sizes (P_B).

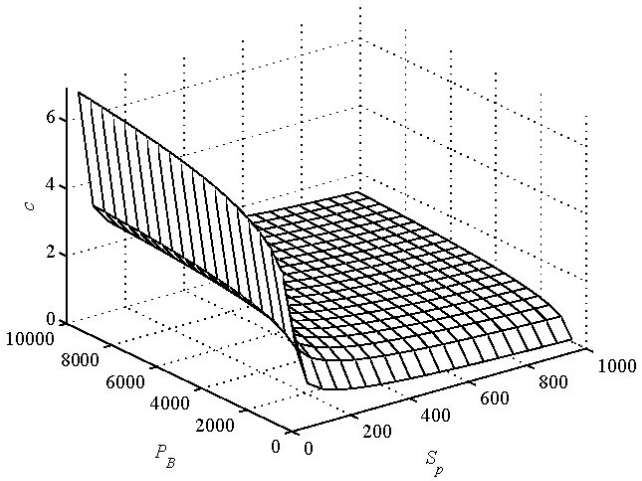

Fig. 6. c value required for our scheme to outperform the EG scheme (c is small compared to S_p and P_S)

4.2.3 Computation Overhead: This is the computation overhead required for comparing the received identifiers of the keys and sub-pools to locally stored keys' material. Initially, a comparison of sub-pools' identifiers is required. If no sub-pools are shared, the protocol is terminated and there is a path key establishment phase . However, if there are one or more shared sub-pools, their internal keys' identifiers are exchanged, and a comparison is performed, until a shared key is obtained. If after comparing all keys' identifiers of shared sub-pools, no shared key is obtained, there is a path key establishement phase. In essence, the computation overhead is probabilistic, based on the probabilities that there is a shared key. The probability that there is no shared sub-pool between two nodes' keying material is $p = p_r$ (number of shared sub-pools =0), which is defined as follows:

$$p = \prod_{i=o}^{S_n-1}\left(1-\frac{S_n-1}{S_p-i}\right) = \prod_{i=o}^{S_n-1}\left(\frac{S_p-i-S_n+1}{S_p-i}\right) \tag{16}$$

The probability that there is no shared keys is $p = p_r$ (number of shared keys = 0) is:

$$p = \left(1-c\frac{S_p}{P_B}\right)^{S_n}\prod_{i=o}^{S_n-1}\left(\frac{S_p-i-S_n+1}{S_p-i}\right) \tag{17}$$

When considering the second scenario of keys selection, where keys are unconditionally selected for each node, the previous probability can be expressed as follows:

$$p = 1-\frac{S_n}{S_p}\sum_{i=0}^{S_P}\frac{c_i}{P_s} = 1-\left(\frac{S_p^{S_n+1}-S_n^{S_n+1}}{S_p^{S_n}(S_p-S_n)}\right) \tag{18}$$

Finally, the average computation overhead is the result of multiplying the absolute overhead for the various aforementioned cases with the corresponding probability for each case.

4.3 Security Assessment

The security assessment of our scheme is in accordance with the same procedure of security analysis as the original work [10]. The main goal of this paper is to introduce a solution for improving the communication efficiency of random key assignment and establishment, rather than a security analysis. In [10], the security has been analyzed in terms of the scheme's resiliency against nodes' compromise. Since we use the same number of keys for each node, the number of keys exposed when a single key is compromised is the same in both cases, and the security level obtained by both schemes is also the same.

5. Special Cases

Our scheme is a generalized form of the EG scheme. The following two cases show this.

1. If we decrease the size of each sub-pool such that each sub-pool includes a single key, the overall number of sub-pools is the same as the number of keys in the large pool, which is equivalent to the EG scheme.
2. When reducing the number of sub-pools to a single sub-pool that includes all the keys in the large pool, the result is equivalent to the EG scheme, where the only sub-pool is selected for each node, and $c = K$ number of keys are selected for it, resulting in global and local connectivity equivalent to the EG scheme.

6. Conclusion

In this paper, we reexamined random key distribution and establishment in sensor networks and extended this work to improve the efficiency and reduce the communication overhead. We also achieved an implicit reduction in the memory requirement for storing identifiers of the keying material. Technically, our contribution relies on the re-design of the large pool, generating smaller pools and assinging randomly generated keys for each node from different and randomly selected sub-pools. The detailed analysis shows that for the same restrictions and resource requirements, with carefully selected parameters, our protocol achieves a greater efficiency with a reduced communication overhead, while achieving the same level of memory overhead for keys and reduced memory requirements for representing their identifiers. Although global connectivity is of concern, we have shown that our scheme achieves the same level of local connectivity, which is more important for the network architecture considered.

References

[1] D. Culler, D. Estrin, and M.B. Srivastava, "Overview of sensor networks," In *Proc. of IEEE Computer Society*, vol. 37, no. 8, pp. 41-49, 2004.

[2] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D.E. Culler, "Enabling sustainable and scalable outdoor wireless sensor network deployments," In *Proc. of the 5th International Conference on Information Processing in Sensor Networks*, pp. 407- 415, 2006.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, Aug. 2002.

[4] A. Wander, N. Gura, H. Eberle, V. Gupta, and S.C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," In *Proc. of 3rd IEEE International Conference on Pervasive Computing and Communications*, pp. 324-328, 2005.
[5] R.J. Watro, D. Kong, S. Cuti, C. Gardiner,C. Lynn, and P. Kruus, "Tinypk: securing sensor networks with public key technology," In *Proc. of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pp. 59-64, 2004.
[6] D.J. Malan, M. Welsh, and M.D. Smith, "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography," In *Proc. of 1st IEEE International Conference on Sensor and Ad Hoc Comm. and Networks*, pp. 71-80, 2004.
[7] D. Nyang and A. Mohaisen, "Cooperative public key authentication protocol in wireless sensor network," In *Proc. of 3rd International Conference of Ubiquitous Intelligence and Computing*, pp. 864-873, 2006.
[8] W. Du, R. Wang, and P. Ning, "An efficient scheme for authenticating public keys in sensor networks," In *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 58-67, 2005.
[9] Y. Maeng, A. Mohaisen, and D. Nyang, "Secret key revocation in sensor networks," In *Proc. of 4th International Conference of Ubiquitous Intelligence and Computing*, pp. 1222-1232, 2007.
[10] L. Eschenauer, and V.D. Gligor, "A key-management scheme for distributed sensor networks," In *Proc. of the 9th ACM Conference on Computer and Communications Security*, pp. 41-47, 2002.
[11] H. Chan, A. Perrig, and D.X. Song, "Random key predistribution schemes for sensor networks," In *Proc. of IEEE Symposium on Security and Privacy*, pp. 197, 2003.
[12] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, and D.E. Culler, "Spins: Security protocols for sensor networks," Wireless Networks, vol. 8, pp. 521-534, 2002.
[13] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," In *Proc. of the 10th ACM Conference on Computer and Communications Security*, pp. 52-61, 2003.
[14] W. Du, J. Deng, Y.S. Han, P.K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *Journal of ACM Transactions on Information and System Security*, vol. 8, no. 2, pp. 228-258, May, 2005.
[15] A. Mohaisen and D. Nyang, "Hierarchical grid-based pairwise key pre-distribution scheme for wireless sensor networks," In *Proc. of 3rd European Conference on Wireless Sensor Networks*, pp. 83-98, 2006.
[16] A. Mohaisen, Y. Maeng, and D. Nyang, "On the grid based key pre-distribution: Toward a better connectivity in wireless sensor networks," In *Proc. of International Workshop on Service, Security and its Data management for Ubiquitous Computing*, pp. 527-537, 2007.
[17] CT (Crossbow technology, wireless sensor networks).
[18] J. Hwang and Y. Kim, "Revisiting random key pre-distribution schemes for wireless sensor networks," In *Proc. of 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 43-52, 2004.
[19] C. Blundo, A.D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," In *CRYPTO'92*, pp. 471-486, 1992.
[20] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks", *Journal of ACM Transactions on Information and System Security*, vol. 8 no. 1, pp. 41-77, 2005.
[21] R. Blom, "An optimal class of symmetric key generation systems," In *Proc. of the EUROCRYPT 84 workshop on Advances in cryptology*: theory and application of cryptographic techniques, Springer-Verlag, pp. 335-338, 1985.
[22] W. Du, J. Deng, Y.S. Han, S. Chen, and P.K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," In *Proc. of 23rd Conference on Computer Communications INFOCOM*, 2004.
[23] S.A. Camtepe and B. Yener "Key distribution mechanisms for wireless sensor networks: a survey," Technical report, Rensselaer Polytechnic Institute, 2005.
[24] T. Ito, H. Ohta, N. Matsuda, and T. Yoneda, "A key pre-distribution scheme for secure sensor networks using probability density function of node deployment," In *Proc. of 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 69-75, 2005.

[25] D. Liu and P. Ning, "Location-based pairwise key establishments for static sensor networks," In *Proc. of 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 72-82, 2003.
[26] R.D. Pietro, L.V. Mancini, A. Mei, "Random key-assignment for secure wireless sensor networks," In *Proc. of 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 62-71, 2003.
[27] R.D. Pietro, L.V. Mancini, and A. Mei, "Efficient and resilient key discovery based on pseudo-random key pre-deployment," In *Proc. of 18th International Parallel & Distributed Processing Symposium*, 2004.

Abedelaziz Mohaisen received a B.Eng. degree in computer engineering from the University of Gaza in 2005, and an M.Eng. degree in information and telecommunications from Inha University, Republic of Korea in 2007. Since October 2007, he has been a member of the engineering staff at Electronics and Telecommunications Research Institute in Korea. He is interested in a wide range of research topics in the areas of information, computer, and networked systems security and privacy.

DaeHun Nyang received a B.Eng. degree in electronic engineering from Korea Advanced Institute of Science and Technology, M.S. and Ph.D. degrees in computer science from Yonsei University, Korea in 1994, 1996, and 2000 respectively. He has been a senior member of the engineering staff at Electronics and Telecommunications Research Institute, Korea, from 2000 to 2003. Since 2003, he has been an assistant professor at the graduate school of Information Technology and Telecommunication at Inha University, Korea where he is also the founding director of the Information Security Research Laboratory. He is also a consultant for Korean Information Security Agency, a member of the board of directors and editorial board of Korean Institute of Information Security and Cryptology. Dr. Nyang's research interests include cryptography and information security, privacy, biometrics and their applications to authentication and public key cryptography. Also, he is interested in the security of WLAN, RFID, WSN, and MANET.

Tamer AbuHmed received a B.Eng. degree in computer engineering from the University of Gaza in 2005. Currently, he is a graduate student in the joint M.Eng./Ph.D. program at Inha University. His research interests are wireless sensor networks security, Internet security, deep packet inspection algorithms and their applications to introduction detection systems. AbuHmed is a student member of IEEE, and IEEE Computer Society.