IEICE TRANS. COMMUN., VOL.E91–B, NO.8 AUGUST 2008

PAPER

Efficient ID-Based Threshold Random Key Pre-Distribution Scheme for Wireless Sensor Networks*

Tran Thanh DAI[†a)], *Nonmember and* **Choong Seon HONG**[†b)], *Member*

SUMMARY Security for wireless sensor networks (WSNs) has become an increasingly serious concern due to the requirement level of applications and hostile deployment areas. To enable secure services, cryptographic keys must be agreed upon by communicating nodes. Unfortunately, due to resource constraints, the key agreement problem in wireless sensor networks has become quite complicated. To tackle this problem, many public-key unrelated proposals which are considered more reasonable in cost than public key based approaches have been proposed so far including random based key pre-distribution schemes. One prominent branch of these proposals is threshold random key pre-distribution schemes. However these schemes still introduce either communication overhead or both communication and computational overheads to resource constrained sensor nodes. Considering this issue, we propose an efficient ID-based threshold random key pre-distribution scheme that not only retains all the highly desirable properties of the schemes including high probability of establishing pairwise keys, tolerance of node compromise but also significantly reduces communication and computational costs of each node. The proposed scheme is validated by a thorough analysis in terms of network resiliency and related overheads. In addition, we also propose a supplementary method to significantly improve the security of pairwise keys established indirectly.

key words: *ID-based, random key pre-distribution, key agreement, threshold, security, wireless sensor networks*

1. Introduction

Wireless sensor networks have attracted academic and industrial researchers' considerable and intensive attention over the past few years due to their promising and wide application in almost every aspect of human life. Some instances of wireless sensor applications include target tracking, environmental monitoring, health care, military surveillance.

In a typical wireless sensor network, the most common communication pattern is the multi-hop wireless communication. Specifically, messages such as routing control information, sensor readings and decisions are transmitted from a source node via multiple consecutive neighboring nodes to a destination node (a base station or cluster head or mobile sink). Due to the openness of wireless medium, inter-node communication is susceptible to simple eavesdropping. In addition, an adversary can make corrupt use of multi-hop communication to mount other malicious attacks such as impersonating, inserting forged or modified or replayed messages into the network to mislead the sensing application. To secure the communication, it is necessary for the communicating nodes to agree on a secret key (pairwise key) for authentication and encryption/decryption in the very first place. The main problem here is how to set up secret keys in sensor nodes so that any pair of communicating nodes can establish a common pairwise key on the fly. This problem is very well established in sensor networks and known as the key agreement problem.

Generally speaking, approaches to the key agreement problem in general network environments can be roughly classified into three categories: trusted server schemes, public key schemes, and key pre-distribution schemes [2]. Nevertheless owing to the challenging features of wireless sensor networks such as large network scale, unknown network topology prior to deployment and constrained system resources, the two first types of schemes have been considered inapplicable or prohibitively expensive to the current generation of sensor nodes. As a consequence, the last few years has witnessed a proliferation of many key pre-distribution schemes for sensor networks [2], [4]–[6], [9], [11]. One prominent branch of these proposals is threshold random key pre-distribution schemes [2], [6]. These schemes exhibit a compelling threshold property: when the number of compromised nodes is less than the threshold, the probability that communications between any additional nodes are compromised is close to zero. This threshold obviously makes it harder for an adversary to obtain small amount of initial payoff from the network via a small number of initial node captures, and makes it necessary for the adversary to attack a large fraction of the network before it can achieve any significant gain. This is a desirable trade-off because small scale network breaches are cheaper to mount and much harder to detect than large scale attacks. Moreover, the schemes also have other nice properties including high probability of establishing pairwise keys, and low storage.

Contributions Motivated by the aforementioned observation, in this paper, we propose an efficient ID-based threshold random key pre-distribution scheme. On the one hand, our scheme is exactly identical to Du et al. scheme [2] and random subset assignment scheme [6] in terms of network resiliency and probability to establish pairwise keys with the same memory cost. The network resiliency here

Manuscript received November 23, 2007.
Manuscript revised March 25, 2008.

[†]The authors are with the Department of Computer Engineering, Kyung Hee University, 1 Seocheon, Giheung, Yongin, Gyeonggi 449-701, Korea.

[*]This research was supported by the MKE under ITRC support program supervised by the IITA (IITA-2008-C1090-0801-0002). Dr. C.S. Hong is the corresponding author.

a) E-mail: trantdai@gmail.com
b) E-mail: cshong@khu.ac.kr

DOI: 10.1093/ietcom/e91–b.8.2602

means that when the number of compromised nodes is less than a threshold, the probability that any nodes except these compromised nodes are security influenced is negligible. This property means that an attacker's gain is decreased for small scale network breach and this gain has a significant security impact only when the attacker mounts a successful attack on a considerable proportion of the network which is considered to be detected easily. On the other hand, our scheme outperforms Du et al. scheme in terms of computational and communication overhead. The performance analysis later shows that our scheme costs less computational overhead than random subset assignment scheme as well. Furthermore, we also propose a supplementary method to significantly improve the security of pairwise keys established indirectly via intermediate nodes in the path-key establishment process.

Organization The rest of this paper is organized as follows: Sect. 2 mentions the related work; Sect. 3 sketches our keystone, the Matsumoto-Imai based scheme; Sect. 4 describes our proposed scheme; Sect. 5 analyzes the resiliency of our scheme against node capture attack and discuss some additional security techniques; Sect. 6 presents the performance analysis; Sect. 7 concludes the paper.

2. Related Work

In this section, we briefly review several noticeable random key pre-distribution schemes for WSNs that have been published in the literature so far.

Eschenauer et al. [4] are the first to propose a random key pre-distribution scheme that relies on probabilistic key sharing among the nodes of a distributed sensor network (DSN). The main idea is that a random pool of keys is selected from the key space. Each sensor node then receives a random key ring from the key pool before deployment. After deployment, any two neighboring nodes able to find a common key within their respective key rings using *shared-key discovery phase* can use that key as their shared secret to initiate communication and to set up the secure connection. In the case that those nodes could not find a common key, they can resort to *path-key establishment phase* to solve the key agreement issue.

Chan et al. [5] further exploited the idea in [4] to develop three mechanisms for key establishment using the framework of pre-distributing a random set of keys to each node. The first one is q-composite keys scheme. The difference between this scheme and [4] is that q common keys, instead of just a single one, are needed to establish secure communication between a pair of nodes. The second one is multi-path key reinforcement scheme applied in conjunction with [4] to yield greatly improved resiliency against node capture attack by trading off some network communication overhead. The third one is random pairwise keys scheme. The purpose of this scheme is to allow node-to-node authentication between communicating nodes.

Du et al. [2] presented a multiple space key pre-distribution scheme for WSNs. This scheme first uses Blom's key generation scheme [8] as a building block to generate multiple key spaces, a pool of tuple (D, G), where matrices D and G are as defined in Blom's scheme. Then this pool is used as a pool of keys as in [4] to establish a common secret key between any pair of nodes.

Liu et al. [6] developed a general framework for establishing pairwise keys between sensor nodes using special bivariate polynomials originated by Blundo et al. [12]. Based on the framework, the authors proposed two efficient schemes: a random subset assignment key pre-distribution scheme and a hypercube-based key pre-distribution scheme.

Chan et al. [12] proposed a variant of random key pre-distribution scheme for key agreement problem in the clustered DSN. This scheme is claimed to increases the resiliency to the attacks on the sensor subgroups.

3. Basic ID-Based Key Pre-Distribution Scheme

In this portion, we present a naïve pairwise key pre-distribution scheme (the *MI-based scheme* for short) motivated by Matsumoto and Imai's proposal [1]. First of all, it is assumed that each sensor node has a unique identification whose range is from 1 to N where N is the maximum number of deployable nodes that could be deployed during the entire lifespan of the sensor network. Each of the unique identifications is represented by $m = \log_2(N)$ bit effective ID in sensor nodes' memory.

3.1 Keying Material Pre-Distribution

A central server first generates l $(m \times m)$ symmetric matrices M_ωs $(\omega = \overline{1, l})$ over finite field $GF(2)$. These M_ωs are kept secret and must not be disclosed to both attackers and sensor nodes. M_ωs is used to generate the ω-th bit of a pairwise key between any pair of neighboring nodes, so l is the length of this key. The central server then computes the keying material for each node S_i as follows:

$$\Phi_i^\omega = y_i M_\omega \quad (\omega = \overline{1, l}) \tag{1}$$

$$\Phi_i = \left[\Phi_i^1\ \Phi_i^2\ \ldots \Phi_i^l\right]^T \tag{2}$$

where y_i $(i = \overline{1, N})$ is the m-dimensional vector, effective ID of node S_i and symbol T denotes transposition operation. Φ_i is the private information and kept secret from both other sensor nodes and attackers. Φ_i^ω and Φ_i are illustrated in the Fig. 1.

3.2 Pairwise Key Establishment

The procedure for establishing a pairwise key between two neighboring sensor nodes S_i and S_j is described as follows with an added optional step to allow explicit key authentication.

Step 1: After being deployed, S_i (S_j) instantly broadcast its effective ID y_i (y_j) to the other node. Since S_i and S_j are neighbors, S_i will certainly get S_j's effective ID y_j and vice

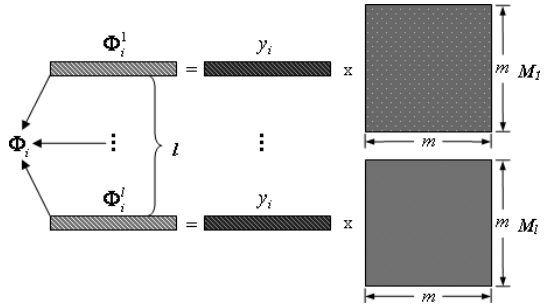

Fig. 1 Structure of sensor node S_i's keying material Φ_i.

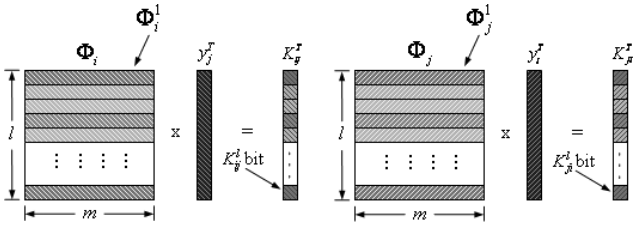

Fig. 2 Pairwise key generating in the manner of the MI based scheme.

versa.

$$S_i \rightarrow S_j : y_i; \quad S_j \rightarrow S_i : y_j$$

Step 2: S_i and S_j use Φ_i and Φ_j respectively to compute their pairwise key as follows:

$$S_i : K_{ij}^{\omega} = \Phi_i^{\omega} y_j^T \quad (\omega = \overline{1, l}), \quad K_{ij}^T = \Phi_i y_j^T \tag{3}$$

$$S_j : K_{ji}^{\omega} = \Phi_j^{\omega} y_i^T \quad (\omega = \overline{1, l}), \quad K_{ji}^T = \Phi_j y_i^T \tag{4}$$

Figure 2 illustrates how the pairwise keys are generated. If y_i and y_j are authentic then $K_{ij} = K_{ji}$ since M_{ω}s are symmetric matrices.

Step 3 (Optional): Up to this step, S_i (S_j) might need to certify that the other has the same key as the one it computed. To do this, S_i (S_j) has to show the other that it has the other's computed key by revealing some secret information without revealing the computed key. Accordingly, S_i (S_j) generates a message, calculates the message authentication code (MAC) of the message as a function of the message and its computed key and then send the message plus MAC to the receiver (MAC can be calculate using a key-dependent one-way hash function such as HMAC).

$$S_i \rightarrow S_j : y_i || y_j, MAC(K_{ij}, y_i || y_j)$$
$$S_j \rightarrow S_i : y_j || y_i, MAC(K_{ji}, y_j || y_i)$$

The recipient performs the same calculation on the received message, using its computed key, to generate a new MAC. The received MAC is compared to the calculated MAC. If the received MAC matches the calculated MAC then the receiver is assured that the message is from the alleged sender and its computed key is exactly the same as that of the alleged sender. Since no one else knows the secret key, no one else could prepare a message with a proper MAC.

Up to this point, any two neighboring sensor nodes have already derived a pairwise key to secure their communication link. However, as shown in [1], our proposed scheme is vulnerable to the *information-theoretic security attack* (a kind of attack based on node capture attack) against the network resiliency. Indeed, our scheme has a certain collusion threshold. As mentioned, M_{ω}s ($\omega = \overline{1, l}$) is a $(m \times m)$ matrix. By using m linearly independent secret Φ_i^{ω}s, M_{ω} can be easily revealed. Therefore, m is the value of the collusion threshold. In other words, an attacker only needs to compromise m sensor nodes to be able to compute any pairwise key of any two uncompromised neighboring sensor nodes using their effective IDs. It implies that with only m compromised sensor nodes, the attacker can compromise the entire network. In the following, an enhanced version of this scheme is proposed to make it more robust against such attack.

4. Efficient ID-Based Threshold Random Key Pre-Distribution Scheme

To enhance network resiliency against node capture attack, we propose an ID-based random key pre-distribution scheme that uses the MI-based scheme as a keystone in combination with the idea of multiple key spaces proposed in [2], [6]. Accordingly, the entire network is depicted in the graph theory language. There is an edge between two neighboring sensor nodes (two vertices in graph theory) if and only if they can establish a pairwise key between themselves. Using the MI-based scheme, a complete graph is guaranteed to create. To obtain the aim of key agreement and enhance resilience, all needed is a connected graph, rather than a complete graph since the latter is a very wasteful use from security point of view.

Some terms need to be clarified before detailing our proposed scheme. We define a key space Ω_i as a 3-tuple (M_i, l, m) of l $(m \times m)$ matrices $M_{i\omega}$, where $M_{i\omega}$ is defined as in the MI-based scheme. A node is said to choose a key space Ω_i if it carries the secret information generated from Ω_i using the MI-based scheme. Two nodes can derive their pairwise key if they have a key space in common.

4.1 Keying Material Pre-Distribution Phase

This phase is performed prior to the sensor node deployment. During this phase, keying material is computed by a central server at a secure location and pre-distributed offline to each node such that after deployment neighboring nodes can derive pairwise keys among themselves using these materials. The security parameters μ, λ and m, where $2 \leq \mu < \lambda$ are also selected. These parameters are chosen with the security and performance in mind which will be discussed later. This phase is performed as follows:

Step 1 (Generating 3-tuples ($\mathbf{M_i, l, m}$)): A central server generates λ key spaces. Each key space Ω_i consists of l $(m \times m)$ symmetric matrices $M_{i\omega}$s as defined in the MI-based scheme.

Step 2 (Generating $\mathbf{\Phi_i}$ matrices): μ distinct key spaces are randomly chosen from λ key spaces for each node. For

each space Ω_i chosen by node S_j, keying material Φ_{ji} is first computed using Eqs. (1), (2) and then stored at this node. Therefore, each node S_j has μ distinct values of Φ_{ji}s. Using the MI-based scheme; two nodes can derive a pairwise key if they have both chosen a common key space.

4.2 Pairwise Key Establishment Phase

After deployment, each node needs to discover whether it shares any key space with its neighbors. To do so, each node instantly broadcasts a message containing the following information: the node's effective ID and the indices of the key spaces it carries.

Suppose that nodes S_i and S_j are neighbors, and then they have received the above-mentioned broadcast messages. If they figure out that they have an identical index of a key space (or identical key space Ω_s), they can easily compute their pairwise key using Eqs. (3) and (4) respectively. Explicit key authentication property can be obtained by having the two nodes participate in the step 3 of the MI-based scheme. However, for fairness, this step will not be taken into account when making performance comparison with other threshold random key pre-distribution schemes. Conversely, there is the case that the two nodes could not establish a pairwise key since they do not have any key space in common. To tackle this problem, there are two possible methods that can be used. The first one has already presented in path-key establishment phase in [2], [4], [6]. The idea is that the two nodes first try to find a secure path protected by newly established pairwise keys and then use this path to negotiate their pairwise key. Assume that the path is $S_i, S_q, \ldots, S_{q+t}, S_j$. To find a common pairwise key, S_i first generates a random key k. Then S_i sends the key to S_q using the secure link between S_i and S_q; S_q sends the key to S_{q+1} using the secure link between S_q and S_{q+1}, and so on until S_j receives the key from S_{q+t}. Nodes S_i and S_j use this secret key k as their pairwise key. Because the key is always forwarded over a secure link, no nodes beyond this path can find out the key.

However this method is vulnerable to node capture attack. Specifically once an attacker can successfully compromise one node on the key path during the key path process, the promising pairwise key k will be disclosed. The node compromise also might lead to a man-in-the-middle attack. Under this attack, a compromised node (man-in-the-middle) S_m forwards a different key k' instead of k onto the secure path to S_j. Consequently, S_m gets control of the communication channel between S_i and S_j since it knows both keys k and k'. Nevertheless, the impacts of the both kinds of attack are very similar and can be diminished by the the second method as mentioned below.

The second one is a combination of the (t, n) secret sharing method [3] and disjoint path finding methods. Accordingly, S_i first discovers the secured disjoint paths to S_j and then uses the secret sharing method to split k into pieces. After that, each piece is sent on one of the secured disjoint paths in the same manner of the first method. Finally, k can be re-constructed if S_j receives no less than t pieces. Obviously this method significantly improves the security of key k since it will not be disclosed if the attack can only compromise less than t nodes. For fairness, when making performance comparison, only the first method is taken into account.

4.3 Selecting μ, λ

The problem here is that given the size and the density of a network, how we can select the values for μ and λ such that the entire network is securely connected with high probability P_{gc}? The approach is that we first compute P_{rlc} (the required probability of two neighboring nodes sharing at least one key space in order to obtain the desired global connectivity P_{gc}). Then we compute P_{alc} (the actual probability of two neighboring nodes sharing at least one key space) using μ and λ. Afterward, the values of μ and λ could be found to satisfy the following inequality

$$P_{alc} \geq P_{rlc} \tag{5}$$

Using the result shown in [4], we can obtain the expected degree of a node d (i.e., the average number of secure links between that node and its neighbors) in order to achieve a given value of P_{gc} when N is large:

$$d = (N-1)\left[\frac{\ln(N) - \ln(-\ln(P_{gc}))}{N}\right] \tag{6}$$

Using a given density of sensor network deployment and wireless connectivity constraints, the expected number of a node's neighbors n can be estimated. Therefore, the value of P_{rlc} can be estimated as:

$$P_{rlc} = \frac{d}{n} \tag{7}$$

Figure 3 illustrates the plot of the estimated P_{rlc} as a function of the network size, N, for various values of P_{gc} and the expected number of a node's neighbors, $n = 40$.

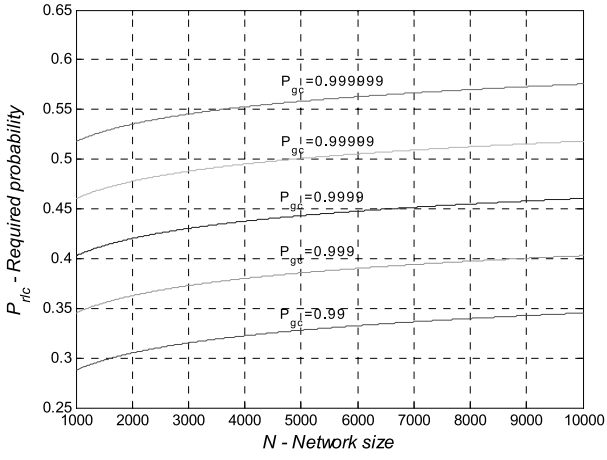

Fig. 3 Estimated P_{rlc} with various N and P_{gc}, $n = 40$.

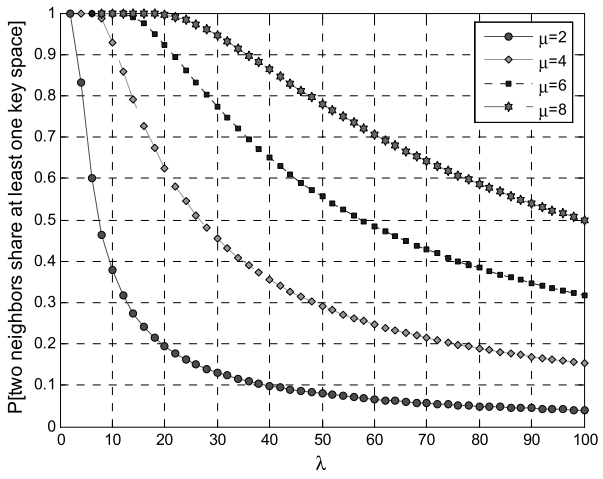

Fig. 4 Probability of finding at least one common key space between two nodes when μ spaces are randomly chosen from λ spaces.

After the values of μ and λ have been selected, the actual probability P_{alc} is determined as follows. Since $P_{alc} = 1 - P$ [two neighbors do not share any key space], we have:

$$P_{alc} = 1 - \frac{\begin{pmatrix} \lambda \\ \mu \end{pmatrix}\begin{pmatrix} \lambda - \mu \\ \mu \end{pmatrix}}{\begin{pmatrix} \lambda \\ \mu \end{pmatrix}^2} = 1 - \frac{((\lambda - \mu)!)^2}{(\lambda - 2\mu)!\lambda!} \tag{8}$$

For better visualization, the values of P_{alc} are plotted in Fig. 4 where λ varies from μ to 100 and $\mu = 2, 4, 6, 8$.

Combining Eqs. (5), (6), (7), and (8), we easily derive the following inequality:

$$1 - \frac{((\lambda - \mu)!)^2}{(\lambda - 2\mu)!\lambda!} \geq (N-1)\left[\frac{\ln(N) - \ln(-\ln(P_{gc}))}{nN}\right] \tag{9}$$

Correspondingly, to obtain a certain P_{gc} of the entire network connectivity with size N and the expected number of neighbors for each node n, all we have to do is selecting values of μ and λ such that inequality (9) is satisfied.

4.4 Addition of New Nodes

Note that our scheme is designed flexibly to accommodate a network of up to for instance $N = 2^{50}$ nodes. Hence addition of new nodes is permitted. In this subsection, issues related to the later node deployment are going to be discussed.

Potentially, there might be two issues. The first one might be involved in the change of value of P_{rlc} and therefore the consistency of the inequality (5). There is a situation that nodes are added later while previous deployed nodes are still alive. In this case, the network density will be increased followed by the increase in the value of n. However, this increase leads to a decrease in the value of P_{rlc} while the value of P_{alc} is unvaried. Thus, the inequality (5) is still correct and the selection of the value of λ and μ from the beginning before the initial network deployment is not influenced. In short, the selection of network parameters from the beginning is independent of the node adding. In the second issue, adding nodes might raise concern about security. Suppose that new added nodes have selected μ key spaces in such a way that all of them contain same information about one key space, for instance Ω_i. The Ω_i will be broken if at least m of these nodes are compromised. However, the probability of Ω_i being broken is less than or equal to $\left(\frac{\mu}{\lambda}\right)^m$ which is close to zero. Consequently, adding nodes does not increase the risk of network security.

5. Security Analyses

In this section, we evaluate our proposed scheme concerning its resiliency against node capture attack. Our evaluation is conducted by finding the answers to two critical questions: (i) Given that b nodes are captured, what is the probability that one key space is broken? To successfully break one key space, an attacker needs to capture at least m nodes that contain this key space's keying material. Hence, the answer to this question quantitatively shows when the network starts to become insecure. (ii) Given that b nodes are captured, what fraction of the additional communications (communications among un-captured nodes) also becomes compromised? The answer to this question shows how much payoff an attacker can obtain after having captured a certain number of nodes.

5.1 Probability of One Key Space Being Broken

Let $\overline{B_i}$ denote the event that key space Ω_i is broken, where $i = \overline{1, \lambda}$, and C_b denote the event that b nodes are captured in the network. Due to the fact that each key space has an equal chance to be broken, then we have:

$$P(\text{one key space is broken } | C_b) = P(B_1 | C_b) \tag{10}$$

Our task now boils down to calculating $P(B_1|C_b)$- the probability of key space Ω_1 being compromised when b nodes are captured. The probability that each compromised node contains information about Ω_1 is $\rho = \frac{\mu}{\lambda}$. Let X denote the number of compromised nodes containing information about Ω_1 after b nodes have been compromised. Then, X is a binomial random variable with parameters (b, ρ). Since the event B_1 can only occur after at least m nodes are compromised, we have the following result:

$$P(B_1|C_b) = \sum_{k=m}^{b} P\{X = k\} = \sum_{k=m}^{b} \begin{pmatrix} b \\ k \end{pmatrix} \rho^k (1-\rho)^{b-k} \tag{11}$$

Combining Eq. (10) and Eq. (11), we derive the following result:

$$P(\text{one space is broken } | C_b) = \sum_{k=m}^{b} \begin{pmatrix} b \\ k \end{pmatrix} \left(\frac{\mu}{\lambda}\right)^k \left(1 - \frac{\mu}{\lambda}\right)^{b-k} \tag{12}$$

Figure 5 shows the relationship between the probability of one key space being compromised, the number of compromised nodes b, the size of key space pool λ, the size of randomly selected key spaces for each node μ, memory usage,

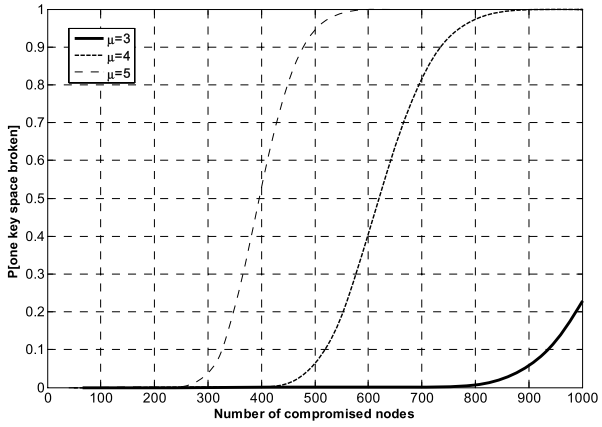

Fig. 5 Probability of one key space being broken given b captured nodes.

and the value of a key space compromise threshold m. For instance, when the threshold m is set to 50, λ is set to 50, and μ is set to 4, then an attacker needs to capture about 500 nodes in order to be able to break one key space with non-zero probability.

5.2 The Faction of Additional Network Communications Being Compromised

To understand how resilient our proposed scheme is, it is better to investigate the influence caused by the event that an attacker has already captured b nodes over the rest of the network. In other words, we like to find out the fraction of additional communications (communications among uncompromised nodes) that an attacker can compromise based on the information obtained from the b captured nodes. In order to evaluate this fraction, what we have to do is to compute the probability that any one of the additional secure communication links is compromised after b nodes have been captured. Note that the additional secure communication links here are the communication links secured by pairwise keys that are established by two uncompromised neighboring nodes.

Let c denote an additional secure communication link, and pk denote the pairwise key used for this link. Let H_i denote the joint event that pk belongs to key space Ω_i ($pk \in \Omega_i$) and space Ω_i is compromised. Hence, we have: $P(c \text{ is broken } |C_b) = P(H_1 \cup H_2 \cup \ldots \cup H_\lambda | C_b)$.

Since c can only use one key and events $H_1, \ldots, H_\lambda$ are mutually exclusive and equally likely. Therefore, we have:

$$P(c \text{ is broken } |C_b) = \sum_{k=1}^{\lambda} P(H_i|C_b) = \lambda P(H_1|C_b)$$

However,

$$P(H_1|C_b) = \frac{P((pk \in \Omega_1) \cap (\Omega_1 \text{ is compromised}) \cap C_b)}{P(C_b)}.$$

Because the event ($pk \in \Omega_1$) is independent of the event C_b or the event (Ω_1 is compromised), we have: $P(H_1|C_b) = \frac{P(pk \in \Omega_1) \cdot P((\Omega_1 \text{ is compromised}) \cap C_b)}{P(C_b)} = P(pk \in \Omega_1) \cdot P(\Omega_1 \text{ is compromised } |C_b)$.

$P(\Omega_1 \text{ is compromised } |C_b)$ is computed by Eq. (11). $P(pk \in \Omega_1)$ is the probability that link c uses a key from key space Ω_1. Since the choice of a key space from λ key spaces is equally likely, we have: $P(pk \in \Omega_1) = \frac{1}{\lambda}$. Therefore,

$$\begin{aligned} P(c \text{ is broken } |C_b) &= \lambda P(H_1|C_b) \\ &= \lambda \cdot \frac{1}{\lambda} \cdot P(B_1|C_b) = \sum_{k=m}^{b} \binom{b}{k} \left(\frac{\mu}{\lambda}\right)^k \left(1-\frac{\mu}{\lambda}\right)^{b-k} \end{aligned} \quad (13)$$

The above equation shows that, given that b nodes are compromised, the fraction of the additional secure communication links compromised is equal to the probability of one key space being compromised.

6. Performance Analyses

In this section, the performance of our proposed scheme in terms of memory usage, communication overhead and computational overhead is analyzed and compared to other threshold random key pre-distribution schemes in [2], [6].

6.1 Memory Usage

For each key space, according to MI-based scheme, each node S_i has to spend $m \times l$ bits on storing the value of Φ_i. Each node also needs a space of $\mu \times \log_2(\lambda)$ bits to store the list of μ key space indices. Thus the total memory usage (bit) MU for each node with μ chosen key spaces is:

$$MU = MU1 + MU2 = m \times \mu \times l + \mu \times \log_2(\lambda) \quad (14)$$

Since the value of m is equal to the value of $\lambda + 1$ in [2] and $t + 1$ in [6]; the value of λ is equivalent to that of ω in [2] and s in [6]; and the value of μ is equivalent to that of τ in [2] and s' in [6], hence memory consumption of our scheme for pairwise key establishment purpose is exactly identical to that of Du et al. [2] and random subset assignment scheme [6].

6.2 Communication Overhead

Note that to establish a pairwise key, the data that each node in our scheme and random subset assignment scheme [6] needs to transmit is its effective ID and the indices of the key spaces in it, while in [2] the data needed to transmit is the node's ID, the indices and the seed of the column of G. Based on that, we draw a comparison as shown in Fig. 6.

Some observations are straightforwardly drawn: (i) Communication overhead of our scheme and random subset assignment scheme are the same. (ii) Choice of an efficient m is quite complicated and bounded by several factors. From Eqs. (12) and (13), the value of m should not be too small otherwise the probabilities of one key space and additional network communications being broken are increased.

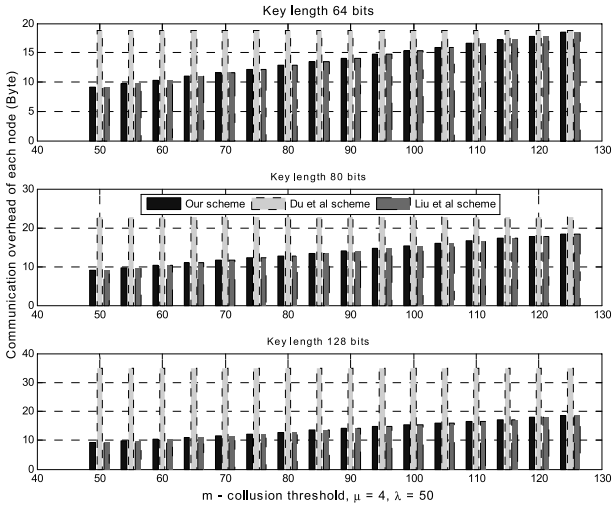

Fig. 6 Extra communication overhead of each node in [2] compared to our scheme.

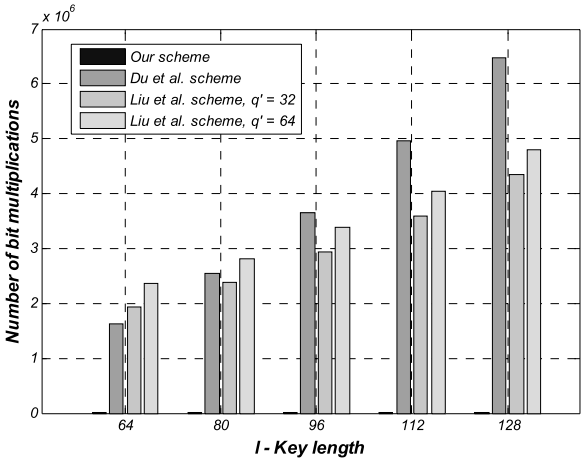

Fig. 7 Computational overhead in each node with various key lengths.

In other words, the entire network resiliency is degraded as the value of m decreases. It should not be too large either since given optimal values of l, μ, and λ, according to (14), the value of m should be as small as possible to lessen the memory overhead. While considering this fact, it is possible that the efficient value of m can be chosen on condition that the inequality $m < 2 \times l$ is assured. In such context, the communication overhead of each node in our scheme is always less than that in [2]. For example, if $m = 50$ (as chosen in [2]) and pairwise key length is 64 bits, then from the figure, the extra communication overhead for each node in [2] in comparison with our scheme is about 10 bytes. It is well known that transmitting a single bit costs as much power as executing 1000 instructions, then the communication overhead of our scheme is far less than that of [2]. (iii) The extra communication overhead of [2] in comparison with our scheme is directly proportional to the length of the pairwise key. Thus, although increasing in key size means an increase in security level but also in communication overhead.

6.3 Computational Overhead

In our scheme, to compute a pairwise key, each node needs to perform a multiplication of a $(l \times m)$ matrix and an $(m \times 1)$ effective ID. Therefore, each node needs $(l \times m)$ bit multiplications. As for Du et al. scheme, the computation cost is due to $2 \times (m - 1)$ modular multiplications. According to Montgomery multiplication algorithm [10], one modular multiplication can be estimated to cost $4 \times l \times (l + 1)$ bit multiplications. Hence, the total computational overhead in Du et al. scheme is estimated as $8 \times (m - 1) \times l \times (l + 1)$ bit multiplications. Roughly speaking, the computation of a pairwise key in random subset assignment scheme [6] is mainly evaluating a t-degree polynomial. This requires t modular multiplications in F_q estimated by $4 \times t \times \log_2 q \times (\log_2 q + 1)$ bit multiplications and $t - 1$ modular exponentiation operations in $F_{q'}$ estimated by $3 \times (t - 1) \times \log_2 q' \times (\log_2 q' + 1) \times (\lceil \log_2(t!) \rceil + t - 1)$. Therefore, the total computational overhead of random subset assignment scheme can roughly estimated by $4 \times t \times \log_2 q \times (\log_2 q + 1) + 3 \times (t - 1) \times \log_2 q' \times (\log_2 q' + 1) \times (\lceil \log_2(t!) \rceil + t - 1)$ bit multiplications. It follows that the computational overhead of our scheme is far less than that in [2], [6]. The numbers in Fig. 7 reinforce our argument.

7. Conclusions

This paper proposes a new key pre-distribution scheme for wireless sensor networks inspired by two types of schemes: ID-based key pre-distribution scheme and random key pre-distribution scheme. Our scheme is scalable and flexible in terms of network size. It not only retains all the highly desirable properties of the existing influential schemes including high probability of establishing pairwise keys, tolerance of node compromise but also significantly improves the security of indirectly established pairwise keys and minimizes communication and computational costs of each node.

References

[1] T. Matsumoto and H. Imai, "On the key predistribution system: A practical solution to the key distribution problem," CRYPTO'87, LNCS, vol.293, pp.185–193, Aug. 1987.
[2] W. Du, J. Deng, Y.S. Han, P.K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," ACM Trans. Info. and Sys. Sec., vol.8, no.2, pp.228–258, May 2005.
[3] A. Shamir, "How to share a secret," Commun. ACM, vol.22, no.11, pp.612–613, Nov. 1979.
[4] L. Eschenauer and V.D. Gligor, "A key-management scheme for distributed sensor networks," Proc. 9th ACM Conference on Computer and Communications Security, pp.41–47, Nov. 2002.
[5] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," Proc. IEEE Symposium on Security and Privacy, pp.197–213, May 2003.
[6] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," ACM Trans. Info. and Sys. Sec., vol.8, no.1, pp.41–77, Feb. 2005.
[7] T.T. Dai, C.T. Hieu, and C.S. Hong, "An efficient ID-based bilinear key predistribution scheme for distributed sensor networks," LNCS, vol 4208, pp.260–269, Sept. 2006.

[8] R. Blom, "An optimal class of symmetric key generation systems," EUROCRYPT'84, LNCS, vol.209, pp.335–338, 1985.
[9] S.P. Chan, R. Poovendran, and M.T. Sun, "A key management scheme in distributed sensor networks using attack probabilities," Proc. IEEE GLOBECOM 2005, pp.1007–1011, 2005.
[10] A. Menezes, P. Van Oorschot, and S. Vanstone, Handbook of applied cryptography, CRC Press, 1996.
[11] W. Du, J. Deng, Y.S. Han, S. Chen, and P. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," Proc. IEEE INFOCOM'04, pp.586–597, March 2004.
[12] C. Blundo, A.D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," CRYPTO'92, LNCS, vol.740, pp.471–486, 1992.

Tran Thanh Dai received the B.Eng. degree in Information Technology from Hanoi University of Technology, Vietnam, in 2005 and M.Eng. in Computer Engineering from Kyung Hee University, Korea, in 2007. Since September 2007, he has been a Ph.D. candidate at Kyung Hee University and working as a research assistant in Networking Lab., Department of Computer Engineering, School of Electronics and Information, Kyung Hee University, South Korea.

Choong Seon Hong received his B.S. and M.S. degrees in electronic engineering from Kyung Hee University, Seoul, Korea, in 1983, 1985, respectively. In 1988 he joined KT, where he worked on Broadband Networks as a member of the technical staff. From September 1993, he joined Keio University, Japan. He received the Ph.D. degree at Keio University in March 1997. He had worked for the Telecommunications Network Lab, KT as a senior member of technical staff and as a director of the networking research team until August 1999. Since September 1999, he has been working as a professor of the School of Electronics and Information, Kyung Hee University. He has served as a Program Committee Member and an Organizing Committee Member for International conferences such as NOMS, IM, APNOMS, E2EMON, CCNC, ADSN, ICPP, DIM, WISA, BcN and TINA. His research interests include ad hoc networks, network security and network management. He is a member of IEEE, IPSJ, KIPS, KICS and KIISE.