A Robust Pairwise Key Predistribution Scheme Resistant to Common Attacks for Wireless Sensor Networks

Tran Thanh Dai[1], Cao Trong Hieu[1],
Md. Mustafizur Rahman[1], and Choong Seon Hong[2]

[1] Department of Computer Engineering, Kyung Hee University
Giheung, Yongin, Gyeonggi, 449-701 Korea
{daitt, hieuct, mustafiz}@networking.khu.ac.kr
[2] Department of Computer Engineering, Kyung Hee University
Giheung, Yongin, Gyeonggi, 449-701 Korea
cshong@khu.ac.kr

Abstract. Security is very necessary for secure operations in wireless sensor networks (WSNs), especially in hostile environments. To achieve security, key agreement needs to be implemented efficiently among communicating nodes. Nonetheless, acquiring such key agreement in WSNs becomes extremely intricate due to resource constraints. Among approaches so far, key predistribution approach is considered as the most suitable solutions. Based on this investigation, in this paper, we propose a pairwise key predistribution scheme relying on sensor nodes' unique authenticated identifications. The identification authenticity is obtained by making use of Merkle hash tree. Our scheme exhibits several attractive properties that are used to resist some common attacks in WSNs as shown in security analysis. We also conduct a detailed investigation of the performance of our scheme in terms of memory usage, communication overhead, and computational overhead.

Keywords: Merkle hash tree, pairwise key, key predistribution, attack, security, wireless sensor networks.

1 Introduction

A wireless sensor network typically consists of a large number of resource limited sensor nodes controlled and managed by one or several powerful control nodes (often called base stations). Sensor nodes are tiny in size and able to sense, process data, and communicate with each other, typically over a wireless media. They are usually deployed in an immense number without any infrastructure support to detect events or physical phenomena, collect and process data, and deliver sensed and processed information to base stations as well as other nodes through its immediate neighbor nodes. Those sensor networks are anticipated to be tremendously applied to various fields of human life ranging from civilian applications to military applications.

When deployed in hostile environments, wireless sensor networks are vulnerable to malicious attacks. In such a context, security assurance becomes one of the major concerns. Typical approaches fulfill security measures using efficient key agreement schemes. Nonetheless, sensor nodes typically operate in unattended conditions; have limited computational capabilities and memory, and battery-power capacity. Due to such resource limitations, the materialization of the efficient key agreement schemes in wireless sensor networks becomes a deeply intricate task. In fact, there are many key agreement schemes proposed for wired and wireless network environments which have been proved to be efficient and secure like trusted server schemes, public key based schemes, and key predistribution schemes. Nevertheless, constrained computation and energy resources of sensor nodes often make the first two schemes infeasible or too expensive for wireless sensor networks [12], [15], [16]. In recent work, there are some attempts to solve the key agreement problem for sensor networks using elliptic curve cryptography (ECC) [17], [18]. However, the energy consumption of ECC is still expensive, especially compared to symmetric key based algorithms. Based on these considerations, it is straightforward to realize that key predistribution schemes seem to be the most feasible solution for the key agreement problem in wireless sensor networks.

One branch of the key predistribution schemes is ID-based, where, no previous communication is required and its key predistribution procedure consists of simple computations. Furthermore, in order to establish the key, each party should input its partner's identifier only into the secret key sharing function [20].

Due to such noteworthy properties, in this paper, we propose a highly resilient, robust, resource-efficient and ID-based key predistribution scheme that provides:

1. Sensor node ID authentication for the pairwise key generation purpose
2. Direct pairwise key establishment with explicit key authentication
3. Substantially improved network resiliency against some common attacks in sensor networks
4. Thorough theoretical analysis of security, memory usage, communication and computation overheads

The remainder of the paper is organized as follows: section 2 mentions the related works; section 3 presents our basic scheme; section 4 describes our security enhanced scheme; section 5 tackles security analysis and common attacks; section 6 presents the performance analysis; section 7 concludes the paper and states our future work.

2 Related Work

Recently, security issues in sensor networks have been received extensive investigation by many researchers. It is already established that due to the limited resources at sensor nodes, asymmetric cryptography based solutions are impratical for sensor networks in most of the application circumstances. In the following, we only mention related work based on symmetric cryptography in a certain form.

Perrig et al. [8] developed a suite of security building blocks named SPINS optimized for resource constrained environments and wireless communication. SPINS is comprised of two secure components: SNEP and μTESLA. SNEP provides the

following security primitives: Data confidentiality, two party data authentication, and data freshness. μTESLA, the second component, is a new protocol which provides authenticated broadcast for stringently resource-limited environments.

Liu et al. improved μTESLA by presenting a new efficient distribution method of key chain commitments for broadcast authentication in distributed sensor networks. Accordingly, in their observation, μTESLA requires initial distribution of certain information based on unicast between the base station and each sensor node before the actual authentication of broadcast messages. This initial unicast-based distribution stringently limits the application of μTESLA because of the limited bandwidth in large sensor networks. To deal with this problem, they proposed a technique to replace the unicast-based initialization with a broadcast-based one that is based on multi-level key chains [9]. They also further extended the ideas in [8], [9] to support a potentially large number of broadcast senders by using a Merkle hash tree [4] to authenticate and distribute the μTESLA parameters [6].

Eschenauer et al. [11] proposed a key predistribution scheme relying on probabilistic key sharing among the nodes of a random graph and uses a simple shared-key discovery protocol for key distribution, revocation and node re-keying. The main idea is to let each sensor node randomly pick a set of keys from a key pool prior to deployment so that any two sensor nodes have a certain probability of sharing at least one common key.

Chan et al. [12] further exploited Eschenauer's idea to developed three mechanisms for key establishment using the framework of predistributing a random set of keys to each node. The first one is q-composite keys scheme. This scheme is mainly based on [17]. The difference between this scheme and [11] 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 multipath key reinforcement scheme applied in conjunction with [11] to yield greatly improved resilience against node capture attacks 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. [15] presented a pairwise key pre-distribution scheme for wireless sensor networks. This scheme first uses Blom's key generation scheme [14] 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 [11] to establish a common secret key between any pair of nodes.

3 The Basic Pairwise Key Predistribution Scheme

Key predistribution schemes are key establishment protocols whereby the resulting established keys are completely determined *a priori* by initial keying material [15]. In this portion, we present a naïve pairwise key predistribution scheme (the basic scheme for short) motivated by Matsumoto and Imai's proposal [1]. Since the Matsumoto-Imai proposal is to apply to the smart-card-based systems, not for wireless sensor networks, we, in the following description, have modified to make the basic scheme suitable for wireless sensor networks. Accordingly, the scheme consists of two

phases, namely *keying material predistribution*, *pairwise key establishment*. The following are detailed description of these phases.

Keying material predistribution. Assume that each sensor node has a unique identification whose range is from 1 to *N* where *N* is the maximum deployable number of sensor 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. The keying material predistribution phase is to predistribute *pairwise key establishing functions* to sensor nodes before deployment such that after deployment, any pair of sensor nodes can find a common pairwise key between them using these functions. It consists of the following steps:

1. Central server generates l $(m \times m)$ symmetric matrices over finite field *GF(2)*. The M^τ s are private information and kept secret from both sensor nodes and adversaries. M^τ is used to generate the τ-th bit of a pairwise key between two sensor nodes, so l is the length of this key.
2. Central server computes the pairwise key establishing function Φ_i for each sensor node S_i by first computing $\Phi_i^\tau = y_i M^\tau$ $(\tau = \overline{1,l})$ and then generating matrix Φ_i as $\Phi_i = \begin{bmatrix} \Phi_i^1 \\ \Phi_i^2 \\ .... \\ \Phi_i^l \end{bmatrix}$ where $y_i (i = \overline{1,N})$ is the *m*-dimensional vector, the effective ID of sensor node S_i. This function is then distributed to each sensor node before node deployment.

Pairwise key establishment. After completing the keying material predistribution phase, each sensor node possesses a pairwise key establishing function. The object of this phase is to establish pairwise keys among sensor nodes using those functions. The procedure for establishing two neighboring sensor nodes S_i and S_j is described as follows with an added step to allow explicit key authentication.

1. After being deployed, S_i (S_j) instantly broadcast its effective ID y_i and a nonce N_i (y_j & N_j) to the other node. N_i and N_j are used here to provide strong message freshness [8]. Since S_i and S_j are neighbors, S_i will certainly get S_j's effective ID y_j and vice versa.

$$S_i \to S_j : y_i \mid N_i$$
$$S_j \to S_i : y_j \mid N_j$$

2. S_i computes the possible pairwise key K_{ij} : $K_{ij}^{\tau} = \Phi_i^{\tau} y_j^T (\tau = \overline{1,l})$ (1), where K_{ij}^{τ} indicates the τ-th bit of the possible pairwise key K_{ij} between S_i and S_j. S_j carries out in the same way to get the possible pairwise key K_{ji}. If y_i and y_j are authentic then $K_{ij} = K_{ji}$ since $M^{\tau} s\ (\tau = \overline{1,l})$ are symmetric matrices.
3. Up to this step, S_i (S_j) needs 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 secret information without revealing the computed key. As in [2], 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 [3]).

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

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

4. 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.

Actually, the basic scheme can be extended so that any pair of nodes within a certain number of hops can establish a pairwise key by having the initiator's effective ID rebroadcast through multi-hops to the receiving end. Through this extension, we can enable end-to-end data protection feature which is considered to be very useful in the hierarchical sensor networks where information (sensed data/command messages) has to be transmitted through multi-hops between a sensing node and a aggregator/cluster head. This feature is significantly energy and computation saving in comparison with hop-by-hop data protection method since encryption and decryption only need to be executed once at the sending and receiving ends respectively. However, in that context, the extended basic scheme also introduces several security flaws. First, it is vulnerable to the man-in-the-middle attack. In addition, it is also an easy target for node replication attack, and node capture attack.

These flaws make the basic scheme not a practical one. In the following, we propose several additional techniques so that we can avoid some attacks and mitigate other attacks.

4 The Security Enhanced Pairwise Key Predistribution Scheme

The essential problem of the basic scheme lies in the fact that it lacks strong node-to-node authentication and node revocation mechanisms. In our enhanced scheme, these mechanisms are proposed based on the Merkle hash tree [4]. Accordingly, we will first build a Merkle hash tree of sensor nodes' IDs (referred as ID authentication tree) and then use this tree in combination with the basic scheme to provide a more security robust scheme against the aforementioned attacks.

4.1 ID Authentication Tree

This tree provide a method for making sensor nodes' IDs publicly available with verifiable authenticity, by using a tree structure in conjunction with a suitable hash function, and authenticating the root value.

Constructing ID Authentication Tree. Without loss of generality, assume the maximum deployable number of sensor nodes (IDs) $N = 2^k$ to accomplish the network mission, where k is an integer. Suppose the central server has a collision-resistant hash function H such as MD5 or SHA1 [25]. To build the tree, for each sensor node $S_i, i \in \{1,...,N\}$, the central server first computes $\varphi_i = H(\Phi_i)$, then computes $I_i = H(y_i \mid \varphi_i)$ - the hash value of the binding between its ID and the hash value of the corresponding pairwise key establishing function. The ID authentication tree is constructed by using $\{I_1,...,I_N\}$ as leaf nodes. Specifically, $I_1,...,I_N$ are arranged as leaf nodes of a full binary tree, and each non-leaf node is computed by applying H to the concatenation of its two children nodes. Figure 1 shows an ID authentication tree of eight sensor nodes' IDs, where $I_i = H(y_i \mid \varphi_i), I_{12} = H(I_1 \mid I_2), I_{14} = H(I_{12} \mid I_{34})$, etc.

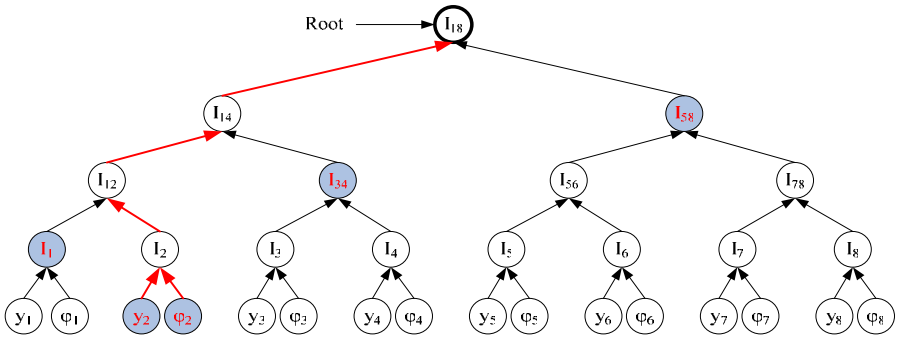

Fig. 1. An example of an ID authentication tree.

Authenticating sensor node IDs. To facilitate the authentication of sensor node IDs, the central server constructs an *ID certificate* for each sensor node. The ID certificate for sensor node S_i consists of its ID and the values corresponding to the siblings of the nodes on the path from the i-th leaf node to the root (the colored one) in the ID authentication tree. For example, the ID certificate for sensor node S_2 in Fig. 1 is $IDCert_2 = \{y_2 \mid \varphi_2, I_1, I_{34}, I_{58}\}$. The central server also pre-distributes the root of the ID authentication tree (i.e., I_{18} in Fig. 1) to sensor nodes.

When a sensor node S_i needs to authenticate itself, it sends a message containing the ID certificate $IDCert_i$. The receiver can instantly authenticate it with the pre-distributed root of the ID authentication tree. For instance, if $IDCert_2 = \{y_2 \mid \varphi_2, I_1, I_{34}, I_{58}\}$ is used, the receiver can instantly authenticate it by verifying whether $H(H(I_{34}|H(H(y_2 \mid \varphi_2)|I_1))|I_{58})$ equals the pre-distributed root value I_{18}. Consequently, the receiver can get the authenticated ID of the sender.

4.2 The Security Enhanced Scheme

This scheme is essentially the modified version of the basic scheme with some additional steps to efficiently resist the common attacks that often occur in the wireless sensor networks as mentioned above. Specifically, it has three phases: *security material predistribution*, *pairwise key establishment* and *pairwise key reinforcement*. These phases are described in more detail as follows.

Security material predistribution. This phase is aimed to provide sensor nodes with security material before deployment consisting of pairwise key establishing functions, ID certificates and the root of the ID authentication tree. It comprises the following steps:

1. Generating symmetric matrices $M^{\tau}s\ (\tau = \overline{1,l})$ as in the basic scheme.
2. Computing pairwise key establishing functions Φ_i s as in the basic scheme.
3. Constructing the ID authentication tree as described in section 4.1.
4. Delivering each ID certificate $IDCert_i$ to the corresponding sensor node S_i in company with the root value of the ID authentication tree.

Pairwise key establishment. The object of this phase is to guarantee two neighboring (non-neighboring) sensor nodes S_i and S_j can securely derive a pairwise key without the interference of man-in-the-middle attack. To achieve this object, S_i and S_j have to conduct the following steps in sequence:

1. S_i and S_j exchange their effective IDs and ID certificates: $IDCert_i$ and $IDCert_j$ through one hop or multihops in the case that one of the two nodes is the aggregator or cluster head.

$$S_i \rightarrow S_j : IDCert_i \mid N_i$$

$$S_j \rightarrow S_i : IDCert_j \mid N_j$$

2. S_i (S_j), after successfully receiving the certificate, verifies the authenticity of the other's ID using this certificate. If the ID is not authentic, the receiver

immediately terminates its communication with the invalid sender. Otherwise S_i and S_j continue with step 3.

3. S_i and S_j compute the possible pairwise keys K_{ij} and K_{ji} respectively using the pairwise key establishing functions as described in the basic scheme.
4. S_i and S_j carry out as in the step 3 of the pairwise key establishment phase of the basic scheme.
5. S_i and S_j carry out as in the step 4 of the pairwise key establishment phase of the basic scheme to get the needed common pairwise key.

After this phase, two nodes can use the established pairwise key to secure their communication link. However, as shown in [1], the proposed scheme is still vulnerable to the information-theoretic security attack (or node capture attack) discussed later against network resiliency. To prevent this sort of attack, two nodes have to participate in the pairwise key reinforcement phase.

Pairwise key reinforcement. This phase is aimed to reinforce a pairwise key between two neighboring sensor nodes S_i and S_j. It happens as follows: S_i and S_j randomly generate k_i and k_j respectively such that their lengths are equal to K_{ij} / K_{ji}. These keys are encrypted and MAC-enabled by K_{ij}, K_{ji} and transmitted to each end. Then, S_i / S_j computes a new pairwise key with S_j / S_i using the formula: $K = K_{ij} \oplus k_i \oplus k_j$ ($K = K_{ji} \oplus k_j \oplus k_i$) (2). In addition to the avoidance of information-theoretic attack, these formulas show that each node has the equal right to decide the value of the potential key *K*. It ensures that no node can get an advantage over the other from *K* selection.

This scheme substantially improves the security robustness of the network. The security robustness as well as performance of the scheme will be thoroughly analyzed in the following sections of this paper.

5 Security Analysis of the Enhanced Scheme

In this portion, we analyze the possible attacks against network resiliency and respective solutions based on the enhanced scheme.

5.1 Man-in-the-middle attack

Attack Scenario. In the multi-level hierarchical wireless sensor networks where there are the need of secure links between sensor nodes and aggregators for communicating the raw information from the sensor nodes to the aggregators and disseminating control/command messages in the reverse direction. In this context, sensor nodes and aggregators can use the basic scheme to establish a pairwise key to secure their communication. However, to establish the key, each party needs to know the other authentic ID. Hence, their IDs have to be exchanged through multi-hops. As a result, the basic scheme is not immune to man-in-the-middle attack. For the sake of presentation, let consider how this attack can be carried out in the following simple scenario as illustrated in Fig. 2.

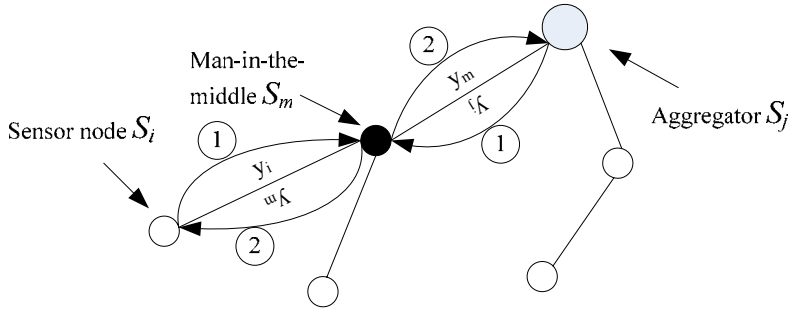

Fig. 2. A simple man-in-the-middle attack scenario in the basic scheme.

1. Sensor node S_i and aggregator S_j exchange their unauthenticated IDs y_i, y_j. Since they are two hops away, the messages have to go through man-in-the-middle S_m.
2. S_m waits for a suitable period of time and then transmits its ID y_m to both S_i and S_j instead of y_j and y_i respectively.

After pairwise key derivation, S_m can successfully eavesdrop on the messages exchanging between S_i and S_j without their detection since S_i has no way other than to believe y_m is the ID of S_j and vice versa.

Resistive Solution. This sort of attack can be defeated using the enhanced scheme with ID authentication (node-to-node authentication) feature enabled.

5.2 Information-theoretic Security Attack

Attack Scenario. Both the basic scheme and the enhanced scheme have a certain collusion threshold. As mentioned earlier, the $M^{\tau}(\tau = \overline{1,l})$ is a $(m \times m)$ matrix. By using m linearly independent secret Φ_i^{τ} s, M^{τ} can be easily revealed. Therefore, m is the value of the collusion threshold. In other words, an adversary 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 adversary can compromise the entire network. Hence, this attack is also one kind of node capture attack.

Resistive Solution. A straightforward solution to the attack is to increase the value of m. However, the increase in the value of m leads to the increase of memory size of sensor nodes needed to store Φ_i. The figure 3 show the relationship between m, pairwise key length l and memory usage. Nevertheless, due to the memory storage constraint of sensor nodes, this solution is not a thorough one.

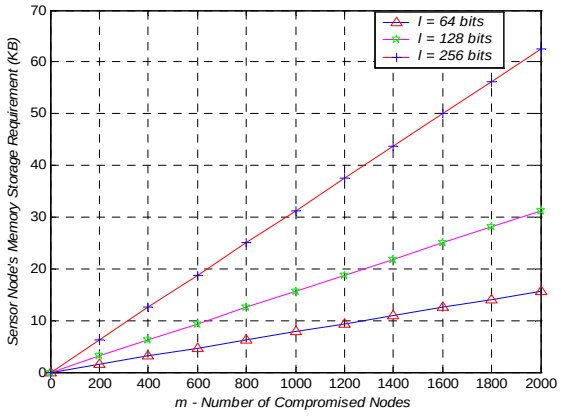

Fig. 3. Memory storage requirement against information-theoretic security attack.

The other solution has been briefly mentioned in the pairwise key reinforcement phase of the enhanced scheme. This solution is partly inspired by an assumption in [5]. Accordingly, in this solution we assume that there exists a lower bound on the time interval T_{min} that is necessary for an adversary to compromise enough m sensor nodes, and that the time T_{est} for newly deployed sensor node to discover its immediate neighbors and establish initial pairwise keys with them is smaller than T_{min}. Taking advantage of the time interval T_{min}, two neighboring sensor nodes need to quickly exchange k_i and k_j to each other and then use (2) to change their initial pairwise key K_{ij} (K_{ji}) to the permanent pairwise key K. By doing in this way, we can eliminate

the *information-theoretic security attack* from the entire network since the adversary could not compute the pairwise key K using (1). Nonetheless, this solution is not scalable enough to protect the sensor nodes that are deployed later on. Since in that context, it might have been enough time for the adversary to compromise enough m sensor nodes and if so the adversary can compute any pairwise key established and used by any newly deployed sensor node despite whether the pairwise key reinforcement is included or not.

To avoid the drawbacks of the two above solutions, we adopt the idea of a revocation tree discussed in [6]. Accordingly, the central server constructs a Merkle revocation tree where the i-th leaf node is the concatenation of ID y_i and r_i where r_i is a random number generated by the central server for each sensor node S_i. The central server then distributes the root of the revocation tree to all sensor nodes. When a sensor node S_j is detected to have been compromised, the central server broadcasts y_j and r_j. To authenticate these values, the central server also has to broadcast the sibling of each node on the authentication path from the leaf node "$y_j \mid r_j$" to the root but not including the root itself. From the broadcast values, any sensor node can recomputed the root hash value, and verify it by checking if it leads to the pre-distributed root value. If the computed root hash value is valid, the sensor node puts the compromised node into a revocation list, and immediately stops terminate its communication with that node. Nevertheless, this solution also has several limitations. The biggest limitation is that it needs an intrusion detection system (IDS) that is beyond the scope of this paper to detect compromised or captured sensor nodes. The other limitations have already been mentioned in [6].

Each solution above has its own certain level of security and cost. Choosing which solution (or combined one of these solutions) to apply depends on particular application of sensor networks and the hostile levels of the deployment area. Thus, there must be trade-offs among security achievements and other sensor nodes' resources.

5.3 Node Replication Attack

Attack Scenario. After compromising sensor nodes, the adversary can replicate the secret information (key establishing functions, and ID certificates) obtained from the compromised nodes in the nodes that under its control and deploy them into the network. These illegal nodes by using the legal secret information can establish trustful relationships with the uncompromised nodes from which the adversary can extract valuable information for further malicious actions.

Resistive Solution. Again we assume that there exists an IDS in the network that can detect malicious behaviors of the illegal nodes. After having the IDS detect the illegal nodes, we use the revocation tree as mentioned in 5.2 to revoke those nodes from the entire network.

6 Performance Analysis of Our Scheme

In this section, we investigate the performance of our enhanced scheme regarding memory usage, communication overhead and computational overhead.

6.1 Memory Usage

As already analyzed in the section 5, in our scheme, memory usage in each sensor node is in proportion to m given pairwise key length l and l given m. Increasing l, m or both result in the increase in security level (key length and collusion threshold) but it implies more memory consumption to store that keying material in sensor nodes (Fig. 3).

In addition, each sensor node also has to spend a certain amount of its memory storage for the purpose of ID authentication. Specifically, each sensor node has to store the root value, and the ID certificate which consists of its effective ID and $h + 1$ hash values, where h is the height of the ID authentication tree. Since the ID authentication tree is a complete binary tree with N leaves, $h = \log N$. Thus, memory usage needed for ID authentication is $L \times (\log N + 2) + m$, where L is the length of a hash value. As a result, the total memory usage altogether is computed by the following equation.

$$MU = m \times (l + 1) + L(\log N + 2)$$

Note that this equation does not include memory usage for pairwise keys and revocation list.

6.2 Communication Overhead

To calculate communication cost, assume that the length of MAC of each message exchanged in the pairwise key establishment phase is μ, the length of each nonce value is ω. Hence, the communication cost is computed by the following equation.

$$\Omega = \underbrace{\underbrace{L(\log N + 1) + m}_{\text{Certificate length}} + \underbrace{\omega}_{\text{Nonce length}}}_{\text{Step 1}} + \underbrace{2 \times m + \omega + \mu}_{\text{Step 4}} + \underbrace{k}_{\text{Last phase}}$$

$$\Leftrightarrow \Omega = L(\log N + 1) + 3 \times m + 2 \times \omega + \mu + k$$

6.3 Computational Overhead

Considering computational overhead of the proposed scheme, it is very straightforward to realize that our scheme is mainly based on multiplications of l vectors between Φ_i^τ s $(\tau = \overline{1, l})$ and sensor nodes' effective IDs over *GF(2)*. These multiplications essentially are exclusion-OR and AND bit operations. These operations consume much less computational time and require much less energy as well. The remaining computations constituting the overall computational overhead are several symmetric key related cryptographic operations. Specifically, each sensor node only needs to execute $1 + \log N$ hash functions to verify the authenticity of each ID certificate. In addition, each node also needs to perform one encryption, one decryption operations in the pairwise key reinforcement phase, two *MAC* generation and two *MAC* verification operations. These operations are considered as the least complex of the cryptographic algorithms and should intuitively incur the least computation and energy cost [7]. Based on these convincing arguments, it is no doubt that the overall computational overhead of our scheme is not worth considering.

7 Conclusion and Future Work

In this paper, we presented a robust pairwise key predistribution scheme resistant to common attacks in wireless sensor networks. Our proposed scheme is a combination of an ID-based key predistribution scheme and an ID authentication method based on Merkle hash tree. Hence, our scheme obviously inherits the merits from that sort of ID-based scheme. First, the number of packets exchanged to establish a pairwise key between two sensor nodes which want to establish a secure communication channel is substantially minimized. Second, the key distribution procedure is composed of simple calculations so that computational costs are quite small and suitable for such computation limited devices as sensor nodes. Finally, each sensor node has only to input its partner's identifier to its secret key sharing function to generate the desired key. By making use of ID certificates to verify the authenticity of sensor node IDs, our scheme is successfully immune to man-in-the-middle attack. Furthermore, our scheme also exposes the resistive solutions to the information-theoretic security attack

(limitation of ID-based key predistribution schemes) and node replication attack. In the later portion, the efficiency of our scheme has been shown by thoroughly investigating its performance in term of memory usage, communication overhead and computational overhead. For these arguments, there is no doubt that our scheme is actually the desired solution to the key agreement problem in wireless sensor networks. In our future work, we intend to develop an Intrusion Detection System which will run in company with our scheme for successful detection of abnormal behavior of compromised nodes and resistive solutions to the other common attacks in wireless sensor networks.

References

1. Matsumoto, T., Imai, H., "On the KEY PREDISTRIBUTION SYSTEM: A Practical Solution to the Key Distribution Problem", Advances in Cryptology - Crypto'87, LNCS vol. 293, pp. 185-193, 1988
2. Stallings, W., "Cryptography and Network Security: Principles and Practice", Prentice Hall, 2nd edn, Jul. 1998
3. Rhee, M. Y, "Internet Security: Cryptographic Principles, Algorithms, and Protocols", John Wiley & Sons, Mar. 2003
4. Merkle, R., "Protocols for public key cryptosystems", Proc. IEEE Symposium on Research in Security and Privacy, Apr. 1980
5. Zhu, S., Setia, S., Jajodia, S., "LEAP: Efficient security mechanisms for large-scale distributed sensor networks", Proc. of the 10th ACM Conference on Computer and Communications Security (CCS'03), pp. 62-72, October 2003
6. Liu, D., Ning, P., Zhu, S., Jajodia, S., "Practical Broadcast Authentication in Sensor Networks", Proc. of the 2nd Annual Inter. Conf. on Mobile and Ubiquitous Systems: Networking and Services - MobiQuitous 2005, pp. 118-129, Jul. 2005
7. Potlapally, N. R., Ravi, S., Raghunathan, A., Jha, N. K., "A Study of the Energy Consumption Characteristics of Cryptographic Algorithms and Security Protocols", IEEE Transactions on Mobile Computing, Vol. 5, No. 2, Feb. 2006
8. Perrig, A., Szewczyk, R., Wen, V., Culler, D., Tygar, D., "SPINS: Security protocols for sensor networks", Proc. of the 7th Annual International Conference on Mobile Computing and Networks, Jul. 2001
9. Liu, D., Ning, P., "Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks", Proc. of the 10th Annual Network and Distributed System Security Symposium, pp. 263-276, Feb. 2003
10. Menezes, A., Van Oorschot, P., and Vanstone, S., "Handbook of Applied Cryptography", CRC Press, Inc., 1996
11. Eschenauer, L., Gligor, V. D., "A key-management scheme for distributed sensor networks", Proc. of the 9th ACM Conference on Computer and Communications Security, pp. 41-47, Nov. 2002
12. Chan, H., Perrig, A., Song, D., "Random key predistribution schemes for sensor networks", Proc. IEEE Symposium on Security and Privacy, pp. 197-213, May 2003
13. Du, W., Deng, J., Han, Y.S., Chen, S., Varshney, P.K., "A key management scheme for wireless sensor networks using deployment knowledge", Proc. of the 23rd AnnualJoint Conference of the IEEE Computer and Communications Societies - INFCOM 2004, Vol. 1, pp. 586-597, Mar. 2004
14. Blom, R., "An optimal class of symmetric key generation systems", Advances in Cryptology - EUROCRYPT '84, LNCS vol. 209, pp. 335-338, 1985

15. Du, W., Deng, J., Han, Y. S., Varshney, P. K., Katz, J., Khalili, A., “A Pairwise Key Predistribution Scheme for Wireless Sensor Networks”, ACM Transactions on Information and System Security, Vol. 8, No. 2, pp. 228-258, May 2005
16. Liu, D., Ning, P., Li, R., “Establishing Pairwise Keys in Distributed Sensor Networks”, ACM Transactions on Information and System Security, Vol. 8. No. 1, pp. 41-77, Feb. 2005
17. Blaß, E.-O., Zitterbart, M., “Towards Acceptable Public-Key Encryption in Sensor Networks”, Proc. of the 2nd International Workshop on Ubiquitous Computing - ACM SIGMIS, May 2005
18. Wander, A. S., Gura, N., Eberle, H., Gupta, V., Shantz, S. C., “Energy analysis of public-key cryptography for wireless sensor networks”, Proc. of the 3rd IEEE International Conference on Pervasive Computing and Communications - PerCom 2005, pp. 324-328, Mar. 2005
19. Blundo, C., Santis, A. D., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M., “Perfectly-secure key distribution for dynamic conferences”, Advances in Cryptology - CRYPT0 '92, LNCS vol. 740, pp. 471-486, 1993
20. Hanaoka, G., Nishioka, T., Zheng, Y., Imai, H., “A Hierarchical Non-interactive Key-Sharing Scheme with Low Memory Size and High Resistance against Collusion Attacks”, The Computer Journal, Vol. 45, No. 3, 2002
21. Wood, A. D., Stankovic, J. A., “Denial of Service in Sensor Networks”, IEEE Computer, Vol. 35(10), pp. 54-62, Oct. 2002
22. Karlof, C., Wagner, D., “Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures”, Proc. of the 1st IEEE International Workshop on Sensor Network Protocols and Applications - SNPA'03, pp. 113-127, May 2003
23. Deng, J., Han, R., Mishra, S., “Security Support for In-Network Processing in Wireless Sensor Networks”, Proc. of the 1st ACM workshop on Security of ad hoc and sensor networks, Oct. 2003
24. Gong, L., David, J., Wheeler, F.R.S., “A Matrix Key Distribution Scheme”, Journal of Cryptology, 1989
25. Du, W., Ronghua, W., Ning, P., “An efficient scheme for authenticating public keys in sensor networks”, Proc. of the 6th ACM international symposium on Mobile ad hoc networking and computing, pp. 58 – 67, May 2005