

A Key Distribution method for Reducing Storage and Supporting High Level Security in the Large-scale WSN

Yoon-Su Jeong^{*1}, Yong-Tae Kim^{*2}, Gil-Cheol Park^{*3}, Sang-Ho Lee^{*4}

^{*1} Department of Computer Science, Chungbuk National University, 410 Sungbong-Ro, Heungduk-Gu, Cheongju 361-763 Korea

^{*2, Corresponding author, *3} School of Information & Multimedia, Hannam University, 133 Ojung-Dong, Daeduk-Gu Daejeon 306-791 Korea

^{*4} School of Electrical and Computer Engineering, Chungbuk National University, 410 Sungbong-Ro, Heungduk-Gu, Cheongju 361-763 Korea

bukmunro@gmail.com, ky7762@hannam.ac.kr, gcpark@hannam.ac.kr, shlee@chungbuk.ac.kr

Abstract

In WSNs, since the attacks, such as jamming or eavesdropping without physical access, easily occur, security is one of the important requirements for WSNs. The key pre-distribution scheme, recently being researched for advances of security in WSNs, distributes keys with probability with the use of q -composite random key pre-distribution method, but there is a high probability that there is no key shared between sensor nodes, and it takes lots of time and energy to find out the shared key, therefore it is not suitable for WSNs. In order to enhance stability of a node that plays a role of gateway without depending on probabilistic key, this paper proposes a key pre-distribution scheme combined with random key pre-distribution scheme and double hash chain. Since the proposed scheme can maintain small storage space and strong security strength, it is more efficient than the existing schemes with the same security strengths. In addition, since it uses a small size of key generation key set, it can reduce a great deal of storage overhead.

Keyword

WSN, Security, Key Pre Distribution, Hash Chain

1. Introduction

Recent development of computer and telecommunication technology has facilitated expansion of WSNs [3, 8]. WSN(Wireless Sensor Networks) means an environment composed of large-scale micro devices which are called sensor nodes [2]. These sensor nodes are characteristic of power supplied by battery, structure with integrated sensor

devices and data processing and short distance wireless communication capabilities. SmartDust and WINS are typical examples for application of sensor network [1].

Application of WSNs includes military sensing and tracking, environmental monitoring, patient monitoring and smart environment, etc. When sensor nodes are installed in a dangerous area, security is very important. For example, attackers can easily peep into traffic, and imitate a network sensor node by giving false information to its neighbor node. For providing WSNs with security, communication should be encrypted and authenticated. This problem can in part be solved by setting a secret key for stable communication between sensor nodes [4].

In the researches conducted till now, sensor node negotiation schemes were proposed, in which key information, which can be used to form a session key, is distributed to all sensor nodes before arrangement, and using this, two sensor nodes generate shared secret keys by themselves after arrangement [1, 2, 3]. As the simplest scheme among them, BROS (BROadcast Session Key Negotiation Protocol) was proposed, in which a shared secret key (master key) is given to all sensor nodes in a sensor network, improving security vulnerability in a session key common to sensor networks, and each sensor node broadcasts a key negotiation message encrypted with the master key, setting a session key with its neighbor nodes [5].

However, the session key forming scheme using a master key has a security problem that exposure of the master key leads to that of all session keys, since the master key is common to all sensors. Therefore, a random key pre-distribution scheme was proposed to solve the problem. The random key pre-distribution scheme has a problem that exposure of a session key may give an influence on security between other nodes

and a vulnerability that there may be no session key shared between two nodes, since the two nodes form a session key through the distribution key common to them and the same session key may be used as the session key of phases of numerous nodes. For improving these vulnerabilities, various schemes were proposed to generate a session key by combining several shared keys [9]. However, these schemes have limits that information on location of sensor nodes is required. OKS (Overlap Key-sharing) was also proposed, in which bit sequence, instead of key, is used to distribute information to each sensor node before arrangement.

This paper proposes a key pre-distribution scheme combining random key pre-distribution technology with double hash chain to enhance energy efficiency and stability of a node that plays a role of key storage space and gateway suitable for WSN environment. The proposed scheme aims at performing random key pre-distribution in which an intermediate node, playing a role of gateway without information on pre-arrangement of network, prevents attacks such as jamming and eavesdropping, using double hash chain. In the proposed scheme, by designing a key pool, the keys are expressed as a key set that generates very small size of keys and are stored into it. The proposed scheme is suitable for a sensor network that uses small amounts of energy, since it requires smaller size of key ring than the existing schemes that require resilience for node capture.

The remainder of this paper is organized as follows: Section 2 presents some works relating to key pre-distribution method in WSNs. Section 3 proposes a key pre-distribution scheme, based on double hash chain for much better resilience property against node capture in WSNs. Section 4 presents performance and security analyses in the proposed scheme. Finally, Section 5 draws conclusions.

2. Related Works

BROSK[5] has advantages that it has small amounts of communication and can form the session key only for each sensor node-pair. It, however, also has disadvantages that the session key of the whole network may be exposed in case of exposure of the master key, since the master key is common to all sensor nodes. Eschenhaur and Gligor proposed a random key pre-distribution protocol that may not heighten security danger, while decreasing storage burden of the sensor nodes generated from the method of the secret key shared between nodes [6]. This protocol is composed of key pre-distribution step before arrangement and common key discovery step after arrangement. A random key pre-distribution

scheme has problems that a session key can be formed only when there exists a key shared between two nodes, and that the same session key can be formed as the session key between lots of nodes.

In OKS (Overlap-Key-Sharing) protocol, a session key is generated by the bit sequence shard by two sensor nodes, using very long bit sequence, instead of a set (P) of a large number of keys [5]. Therefore, it allows the sensor nodes to randomly assign and store bit sequences which are part of long bit sequences in network. Each sensor node broadcasts information on its own stored bit sequences and receives information on the bit sequences broadcast by its neighbor node, thereby comparing it with its own stored bit sequences. Through a hash function, it then forms a certain size of session key with the bit sequences of an overlapped interval shared by its neighbor node. This enables it possible to save the amounts of storage and communication, compared with a random pre-distribution scheme, but there exists a disadvantage of decrease in the possibility to connect a session key between two sensors.

Q complex key scheme, where a session key is generated only when there exists more than a certain number (q) of session keys shared by two nodes, is composed of key pre-distribution step and shared key development step [7]. In the key pre-distribution step, its own node ID is broadcast, like in a random key pre-distribution protocol, and its neighbor (within the range of communication) node ID is checked. In the shared key development step, each sensor node transmits its neighbor nodes the puzzle (client Merkle puzzle) for m keys within its own key ring. When the neighbor node receives the puzzle, it finds out the key that can provide the right answer for the puzzle (that is, the key shared by the two sensor nodes) within its own key ring, and transmits the right answer to the transmitting sensor. When there is more than a certain number (q) of keys shared with its neighbor node, a session key between the two node is formed from the shared key through hash. Q complex key scheme enhanced stability by lowering the possibility that different sensor node-pairs have the same session key, and prepared for eavesdropping attack, using a puzzle.

However, since it transmits each puzzle as a separate message, preparing for replay attack, it has the problem that the amounts of transmission, which can give a great influence on the most vulnerable electric power of a sensor node, is excessively increased, and that mutual confirmation process for the formed key is omitted.

3. The Key Pre-distribution Using Double Hash Chain in WSN

This session describes a random key pre-distribution scheme based on double hash chain to explain the system model and provide resilience for node capture in a sensor network.

3.1 Notations

This session describes notation used in the proposed scheme like Table 1 below.

Table 1. Notation and assumption

Notation	Description
x, y	Two generic sensors in the WSN
N	Number of sensors in the WSN
S	Size of the pool from which the keys are drawn
K_i	i^{th} key pool assigned to each sensor
$C_{x,y}$	Random number of x,y
g_i	Unique generating key
$E()$	Encryption function
$H()$	Hash function
\parallel	XOR operation

3.2 The Key Pre-distribution Scheme

This section describes a random key pre-distribution scheme for stable communication of an intermediate node that plays a role of gateway. The key pre-distribution scheme is largely composed of key pre-distribution, shared key restoration and path key establishment syntaxes. In the key pre-distribution syntax, a large key pool and ID of a node is generated before arrangement of network. Each node assigns m key rings and randomly take them out of the pool. In the shared key restoration syntax that is used during network setup, all nodes broadcast ID of the key existing in its key ring. A node can find out its neighbor node sharing a key through such broadcasting process. Finally, in the path key establishment syntax, the path key is configured with ID and key-pair of a node for secure communication with its neighbor node. In this section, our scheme consists of key pre-distribution, link key setup within each cluster and path key establishment.

3.2.1 Key Pre-distribution

In the key pre-distribution syntax, all nodes are divided into many groups, and at least n nodes are

configured in each group. All sensor nodes, included in the same group G_i , perform prior loading of bootstrapping program that controls role division and node configuration. The sensors, composing group G_i , have key pool K_i of group G_i . Key pool K_i is composed of size of key pool S and number of key chain n . The sensor node performs legitimate authentication, using a signature key, instead of cluster. A malicious node cannot fake a key of sensor node, resulting in the advantage that it can be used in various environment services through this process. Key pool \mathbf{K} is composed of different key chain \mathbf{L} . K_i is composed of $\mathbf{K} = c_{j,i} \cap c_{j,n-i+1}$ ($i=n-1, n-2, \dots, 0$, $j=1, 2, \dots, n$) and $c_i \cap c_{i+1} = \phi$ ($i \neq j$). Each key chain K_i is generated through g_i , the only generation key, and seed is obtained through repeated application of the keyed hash algorithm. The l^{th} key of the key chain \mathbf{L} is computed with the use of random number $c_{j,n} \cdot g_{j,n}$ of $h_{c_{j,n} * g_{j,n}}^n$ is secret information values strictly maintained with other nodes in WSN. Generates signature key $S((c_{j,0})_{CH-S}, SK_S)$, using the key shared between cluster head and sensor node.

3.2.2 Link Key Setup within Each Cluster

The key establishment syntax is operated, based on double hash chain. The key of j^{th} hash chain has value K^{j-1} . When authentication is required by a sensor node, the cluster head checks the key of the sensor node. When double hash chain is generated, one chain is generated by a sensor node, and the other chain is generated by the cluster head. The chain generated by one chain of the double hash chain is composed of pairs of hash function value $c_{j,n-i+1}$ in mutual exchange order of numbers. A sensor node randomly selects seed value $c_{j,n}$ and apply it to the hash function, generating other chain. Figure 1 shows the initial step generating double hash chain, more detailed operation process of which are as follows:

Step 1: When authentication is required by a sensor node, the cluster head checks information of the sensor node. If the information is correct, a key is given to the sensor node; if not, the process finishes.

Step 2: The sensor node generates double hash chain so that the nominal amounts may be differentiated by number of hash chains j . In order to generate one hash chain, the sensor node selects random number $c_{j,n}$.

The sensor node apply the selected $c_{j,n}$ to the hash function, generating one chain of double hash chain like $h(c_{j,n+1}, g_{j,n+1})$, ($n=n-1, n-2, \dots, 0$). Root value $c_{j,0} = h^n(c_{j,n})$ is generated when the hash function is applied to $c_{j,n}$ for n times.

Step 3: In order to generate the other chain of double hash chain, the cluster head selects random number $c'_{j,n}$, using $h^n(c_{j,n}, g_{j,n})$. In order to apply $c'_{j,n}$ selected for n times, $c'_{j,0} = h^n(c'_{j,n})$ is computed. Seed value of the generated hash chain is encrypted $K_{node-cluster}$ with key of the sensor node and given to the sensor node.

Step 4: The cluster head sends $R_j = S(n \cdot Root_j, SK_C)$ to sensor nodes. The value marked on $Root_j$ is given by the sensor nodes. When keys are divided in the cluster, the proxy signature key-pairs of BS (Base Station), instead of the BS, are sent to the cluster head. When the cluster head uses the proxy signature key-pairs, fairness of the cluster head for the keys is verified.

3.2.3 Path Key Establishment

In network bootstrapping syntax, each sensor node broadcasts key index information of key ring R_j to obtain key information of its neighbor node. This enables each node to know the key of its neighbor node. Then each node surveys key index information of its own key ring to compute or find out a key shared with its neighbor node. The sensor node selects $c_{j-1,0} = c_{j,n} \cdot s_{BS}$ and random number, apply them to the hash function for n times. The sensor node agrees on indication for a new key, using proxy key signature of BS. Sensor node A sends $S(c_{j-1,0}, c'_{c-1,0}, j-1, n, R_{j-1}, r_{BS}, SK_S)$ and $Cert_S$ to sensor node B included in other cluster. Sensor node B checks $Cert_S$ and verifies $V(Root_{j-1}, R_{j-1}, r_{BS})$, using the shared key of BS. After verification of $V(Root_{j-1}, R_{j-1}, r_{BS})$, sensor node B delivers the generated key to sensor node A, resulting in mutual communication.

4. Analysis

This session evaluates the relation between key ring size and security strength of a sensor node in a given key sharing probability P_m . This session

assumes that each sensor node made security negotiation with its neighbor node in initial step.

4.1 Performance Analysis

Numerical evaluation of performance analysis was preformed with the use of Matlab. For simplicity of the proposed scheme, we assume that examples of all group key pre-distributions have the same property of functions. This assumption is as same as that of all key pre-distribution scheme of given [4] in the same storage space, group size and keying material size.

We evaluate the proposed scheme with the storage space required for a sensor node in a given key sharing probability P_m . It is assumed that two specific node n_i and n_j share at least one key and q keys. For key pre-distribution scheme, P_m is computed like $1 - (1 - \frac{1}{s})^m$. Here s is the size of key ring, and m is the size of key pool. \hat{d} counts the number of sets assigned to the key pool. Probability \hat{d} is as same as j that obtains m from the key pool and satisfies $1 \leq j \leq d$. The number of sets in the key pool is expected as $1 + (d-1)P_m$. When the key ring is assigned to other node, probability \hat{d} is as same as j that obtains m from the key pool and satisfies $0 \leq j \leq d$. On the basis of above equations, the expected number of key pool setup can be considered as \hat{d} .

Performance analysis of the proposed scheme can be expressed as various values of K , L and (r_0, r_1) pairs. In a given large number of K and L , we can observe better property of node capture. For example, in comparison of security length, that of 210 keys in the proposed scheme is shorter than that of 100 keys with $K=100,000$ in Eschenauer scheme. Although the proposed scheme is worse than Eschenauer scheme due to shorter security length, its R value is over 30 less than that of Eschenauer scheme. When security length is guaranteed like this, the key ring size, required in the proposed scheme, is about 20% less than that of Eschenauer scheme. This means that the proposed scheme has more enhanced performance than other schemes when network size increases byth of the proposed scheme with $n=10,000$, $P_m=0.5$ and key ring=256.

The results of Figure 1 show that the proposed scheme has 28% lower values than Eschenauer scheme in the same environment when 100% of negotiated communication fraction is performed. As a result, this enables to obtain more efficient storage space in the

same security strength, since it is possible to obtain lower values in increase of network size.

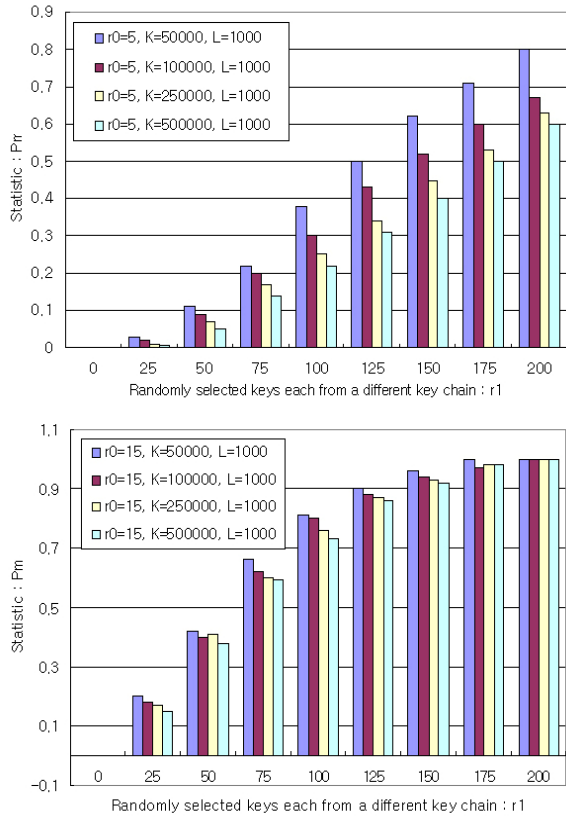


Figure 1. p vs. r_0 and r_1 under different values of $K=\{50,000, 100,000, 250,000, 500,000\}$

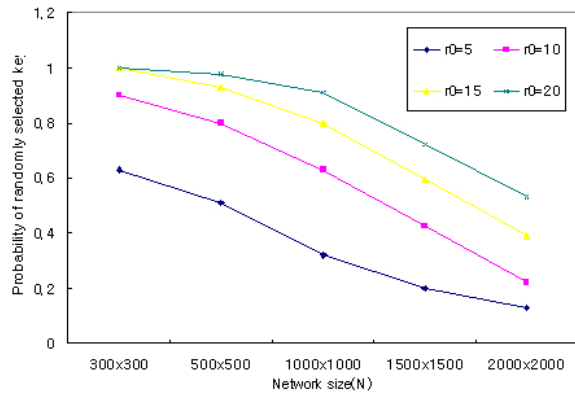


Figure 2. Probability p_m of randomly selected keys in areas of various sizes

Figure 2 shows probability p_m of randomly selected keys in areas of various sizes of network. Here

we can know that probability of randomly selected keys adversely increases with area size. This result can be obtained from the value line drawn through value analysis of different K , L and (r_0, r_1) pairs. In Figure 3, probability of randomly selected keys is in proportion to $\frac{2}{n}$.

4.2 Security Analysis

In the proposed scheme, the hash function, which generates a key, is satisfied only on the aspect of key generation. This is because it is possible to generate a key from seed value or verify it, but is impossible to compute it. If $(c_i, c_{n-1}), \dots, (c_{i+k}, \dots, c_{n-i-k})$ is used for authentication, a third person cannot generate hash value of c_i for $j > i+k$. This makes it impossible to perform key fake.

Moreover, the proposed scheme, which proposes use of double hash function, is more secure in key configuration than the existing schemes which use one hash function. If validity of sensor is to be proved, the divided key should be authenticated after it is generated as computation of $c'_{j-1} = c'_{j,n}$, the seed value of chain. If a sensor node did not setup seed value of the key in order to be divided into $c'_{j,n} \cdot s_{BS}$, the cluster head can determine validity of the sensor by checking seed value of the sensor or verifying validity of division. BS, which has information of sensor node for authentication, can track sensor node's generation of faked key.

When a sensor node makes double use of $(c_{j,j+1}, c'_{j,n-i})$, BS searches for the sensor node that generated a key after computing root value of the key. Then, for one key chain used for key division, it is possible to compute seed value of the key used before, using the root value of BS, thereby searching for the sensor node that generated the key. If the sensor node divided the key and reused the divided key, BS makes seed value of the generated key, using the divided and generated key, and tracks the sensor node, using sensor node information stored when proxy signature was given.

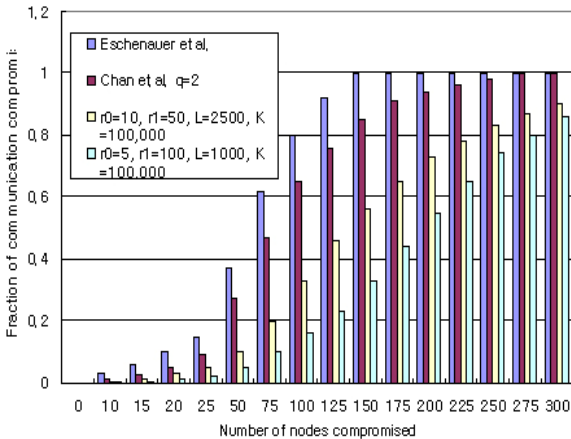


Figure 3. Security strength of the proposed scheme with $n=10,000$, $p_m=0.5$, round=256

Figure 3 shows security strength of the proposed scheme with $n=10000$, $p_m=0.5$ and key ring size $R=256$. Figure 4 shows that the proposed scheme has 38% more efficiency than Eschenauer scheme in the same fraction of communication compromised in 100% performance of Eschenauer scheme. In addition, results of Figure 4 show that the proposed scheme is more efficient than the scheme of Chan et. al that has good security strength only for small size of attack, since the proposed scheme has good security strength, irrespective of network size.

5. Conclusion

Since jamming or eavesdropping can more easily occur in wireless network than wire network, security is one of the most important factors in WSNs. Since lots of keys have to be loaded for operation of each node in the existing key pre-distribution schemes, they are not suitable for large-scale sensor networks. This paper proposed a random key pre-distribution scheme that does not give an influence on network size. In order to decrease storage space, while maintaining the same security length, instead of assignment of all keys to the sensor nodes, the proposed scheme maximized efficiency by saving important key values into key generation set, using key pool. In particular, the proposed scheme showed average 13% enhancement of security strength length, compared with the schemes of Eschenauer and Chan et. al. Further study in the future is planning to evaluate security strength, according to the kinds of active attacks in case of random node capture in an optimized size.

6. References

- [1] J. D. Richard and S. Mishra, "Security Support for In-Network Processing in Wireless Sensor Network", Proc. of ACM Workshop on SASN, pp. 83-93, 2003.
- [2] S. Doshi and A. Eswaran, "A Hierarchical Security Architecture for Group Communication in Sensor Network", Project Report, 2003.
- [3] S. Slijepcevic, M. Potkonjak, V. Tsiatsis, S. Zimbeck and M. B. Srivastava, "On Communication Security in Wireless Ad-Hoc Sensor Network", Proc. of WETICE, pp. 139-144, 2002.
- [4] H. Chan and A. Perrig, "PIKE: Peer intermediaries for key establishment in sensor networks", Proc. of IEEE INFOCOM, 2005.
- [5] B. Lai, D. Hwang, S. Kim and I. Verbauwhede, "Reducing Radio Energy Consumption of Key Management protocols for Wireless Sensor Networks", Proc. of ISLPED'04, pp. 351~356, 2004.
- [6] L. Eschenhour and V. Gligor, "A Key-Management Scheme for Distributed Sensor Networks", Proc. of CCS'02, pp.41~47, 2002.
- [7] H. Chan, A. Perrig and D. Song, "Random Key Predistribution Schemes for Sensor Networks", Proc. of 2003 IEEE Symposium on Security and Privacy(SP'03), p. 197~213, 2003.
- [8] R. Pietro, L. Mancini and A. Mei, "Random Key Assignment for Secure Wireless Sensor Networks", Proc of 1st Workshop Security of Ad Hoc and Sensor Networks, pp. 62~71, 2003.
- [9] R. Blom, "An Optimal class of symmetric key generation systems", Proc. of EUROCRYPT84, Lecture Notes in Computer Science, Springer-Verlag 209, pp. 335~338, 1984.