A KEY-EXCHANGING SCHEME FOR DISTRIBUTED SENSOR NETWORKS

Hung Le Xuan, Sungyoung Lee, and Young-Koo Lee
Department of Computer Engineering,
KyungHee University, Korea
lxhung@oslab.khu.ac.kr, {sylee, yklee}@khu.ac.kr

Abstract: In order to achieve secure node-to-node communication in the distributed sensor networks, key management is the most important issue. However, due to limitations of sensor nodes in terms of energy, storage and communication bandwidth, this is a non-trivial job. Pre-distribution of secret keys is one of the most efficient ways. Currently, several key pre-distribution schemes for distributed sensor networks have been proposed. To our best knowledge, all of these efforts are on how to distribute shared keys in efficient manner. However, they do not consider the efficient path of node-to-node communication. In this paper, we propose a additional key-exchanging scheme for existing key pre-distribution scheme. We show that, by shifting secret keys to neighboring nodes, we reduce communication and computation overhead noticeably for the networks.

1. INTRODUCTION

Nowadays, distributed sensor networks (DSNs) are one of the most emerging technologies. Many applications such as distributed information gathering and distributed micro sensing in radiology, military, and manufacturing drive the research in sensor networks. However, DSNs differ from the traditional ad hoc network in several important areas. As consequently, security schemes for ad hoc networks can not be applied to DSNs such as public key scheme of Diffie-Hellman [3] or singly mission key.

One of the most important challenges of sensor network security is the design of protocols to bootstrap the establishment of a secure communication infrastructure, i.e. establishing a common key between two nodes so that they can communicate with each other in secure manner. The difficulty of the bootstrapping problem stems from the numerous limitation of sensor networks. In order to solve this problem, there have been three types of bootstrapping schemes: trusted server scheme, self-enforcing scheme, and key pre-distribution scheme [2]. The trusted-sever scheme depends on a trusted server for key agreement between two nodes, e.g. Kerberros [4]. This type of scheme is not suitable for sensor networks since there is usually no trusted node in sensor networks. The self-enforcing scheme depends on asymmetric cryptography. However, this scheme is unfeasible for sensor networks due to energy and memory limitation of sensor nodes. The other scheme, key pre-distribution seems to be the most suitable. In this scheme, key information is distributed among all sensor nodes prior to deployment.

Eschenauer and Gligor [1] recently proposed a random key pre-distribution scheme to address the bootstrapping problem. In this paper, we refer this scheme as the basic key pre-distribution scheme that we would like to improve performance. The operation of this scheme is briefly described in section 2. However, the problem of this scheme is that after path-key establishment phase, when two sensor nodes would like to transmit data through secure link, the message may travel along a long path before reach the destination. We name this as long-way exhaust problem of all key pre-distribution schemes. In order to solve this problem, we propose an additional key-exchanging phase to the basic scheme. The main idea is that after path-key establishment phase, each node broadcasts a notification message through the entire the network looking for the key with its neighbors. If such a node exists, it shifts the key to the broadcasting node along a secure path. After key-exchanging phase, every node shares at least one common key with each of its neighboring nodes. By shifting common key to two neighboring nodes, every node can find a shortest route to another node in the sensor networks.

The remaining paper is organized as follows. We first present an overview of the basic scheme in Section 2. Section 3 describes our key-exchanging scheme. We analysis our scheme and compare to the basic scheme in Section 4. We also give some discussion in Section 5. Section 6 concludes the paper and figures out some issues for our future work.

2. BASIC RANDOM KEY PRE-DISTRIBUTION SCHEME.

In [1], Eschenauer and Gligor proposed a Random Key Pre-distribution scheme based on probability model. This scheme is including three phase: key-predistribution, shared-key discovery, and path-key establishment.

Key Pre-distribution phase is processed before network deployment. A key pool S is created with keys. Each node randomly picks m keys from this pool and stores them in its memory. This set of m keys is called the node's key ring. The number of keys in the key pool, |S|, is chosen such that two random subsets of size m in S will share at least one key with some probability p.

After the sensor nodes are deployed, a key-setup phase is performed. During this phase, each node attempts to find out which node it shares a key with. To do this, every key is assigned with short identifier prior to deployment, and each node broadcasts this set of identifiers. If such a key exists, the key is used to secure the communication between these two nodes.

After key-setup is complete, a connect graph of secure link is established. Nodes can then setup path keys with their neighbors with whom they do not share keys. If the graph is connected, a path can always be found from a source node to any of its neighbors. The source node can then generate a path key and send it secure via the path the target node.

In this scheme, the authors attempt to provide high connectivity with less required memory, regardless to efficient communication later on. Assume that there are two neighboring nodes communicate with each other but they do not have any shared key. According to this scheme, in order to guarantee secure communication between these nodes, packets must be sent through path-keys which have been formed. It's obvious that such path is usually a long-way communication and consumes much energy of sensor nodes. How can we shorten this path while still guarantee secure communication between end-to-end nodes communication? In this paper, we solve this problem by using an additional phase, the key-exchanging phase.

3. A KEY-EXCHANGING SCHEME

In the basic scheme, any two neighboring nodes need to find a path-key in order to establish a secure link to transmit their packets. These paths are usually not efficient for routing protocol in terms of energy consumption and end-to-end delay. Thus, we propose a modification to the basic scheme where key-exchanging phase is additionally performed after path-key establishment. By shifting keys to neighboring nodes, we significantly increase the energy efficiency while still maintain original security of the basic scheme.

Figure 1 describes a simple case of long-way exhaust problem of the basic scheme. Assuming that after path-key establishment phase, network graph connection is presented as Figure 1. Considering that two nodes A and H would like to communicate to transmit packets via a secure communication. Thus, node A first has to send to node G which shares a common key . Node G then forwards to node E by encrypting the message with shared-key , so on and so forth until the message reaches the destination node H. As the results, the message travels along the path A-G-E-F-H. This long way costs much communicational and computational cost of sensor nodes for transmission, reception, key verification, message encryption, etc. Our approach solves this problem by establishing a secure link between G and H so that A can find the shortest path to H. In other words, our approach supports every node to find the shortest path to the destination.

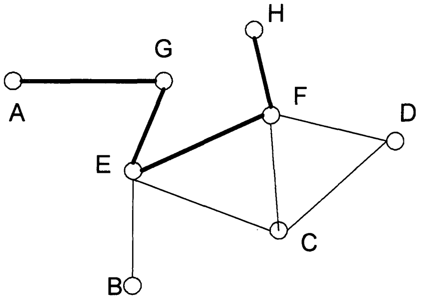

Figure 1. A Long-way exhaust problem of the basic scheme

3.1 Description of the Key-Exchanging Scheme

Notation	*Description*
n_i	Sensor node *i*
id_i	Identifier of node *i*
nonce	Random nonce value
K_{AB}	Private Key shared between A and B ($K_{AB} = K_{BA}$)
$E(K, M)$	Encryption message *M* with key *K*
$\|\|$	Concatenation operation

Table 1. Notation used in Key-Exchanging Scheme

The operation of the key-exchanging scheme is similar to that of the basic scheme, different only in the additional phase as illustrated in Fig.2. In this operation flow, key-exchanging phase is performed after path-key establishment phase. Table 1 describes the notation used in our scheme.

3.2 Key-Exchanging Phase (Additional step)

The protocol for the *key-exchanging phase* is as follows:

$$n_i \rightarrow broadcast \;\; id_i \,||\, \{id\}_{i'}$$

Firstly, each node n_i includes its identifier id_i along with all identifiers of its neighboring nodes $\{id\}_{i'}$ in the "*hello*" it broadcasts after path-key establishment. In order to simplify the computational and communicational cost, message is transmitted without any encryption.

$$n_i \leftarrow n_j \quad id_{n_j} \,||\, E(K_{(\gamma-1)\gamma}, K_{ji'} \,||\, nonce)$$

We assume that node n_j posses a shared key $K_{ji'}$ with one of n_i's neighboring nodes, say $n_{i'}$. n_j then replies to n_i along a secure path $\Gamma = \{n_j, v_1, v_2, ..., v_l, n_i\}$ in sequent order[1]. Here, $K_{(\gamma-1)\gamma}$ is a secret key between two neighboring nodes $x_{\gamma-1}$ and x_γ on the secure path Γ, i.e. $(x_{\gamma-1}, x_{\gamma-1}) \in \{(n_j, v_1), (v_1, v_2), ..., (v_l, n_i)\}$. n_j then marks $K_{ji'}$ as a *exchanged-key*. This is important since after *key-exchanging phase*, every node should remove all *exchanged-keys* to release their memory.

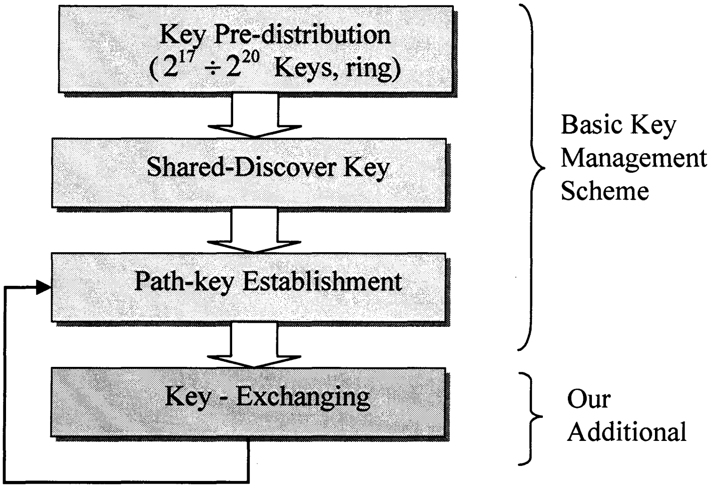

Fig. 1. Work Flow of Key-Exchanging Scheme

[1] To make key-exchanging more secure, n_j may disassembles $K_{ji'}$ into a set of $K_{ji'} = \delta_1 \oplus \delta_2 \oplus ... \oplus \delta_n$ and sends each number δ_i through different secure paths. When n_i receives all $\{\delta_1, \delta_2, ..., \delta_n\}$, it infers the key $K_{ji'}$ by assembling $K_{ji'} = \delta_1 \oplus \delta_2 \oplus ... \oplus \delta_n$.

After key-exchanging phase, path-key establishment phase is performed again. The purpose of this repeated step is to clear out all unnecessary path-key and setup a new path-key for the entire network.

4. ANALYSIS

In the basic scheme [1], Eschenaeur and Gligor used Random Graph theory to analyze DSN connectivity. A random graph *G(n,p)* is a graph of n nodes for which the probability that a link exists between two nodes is p. In a large sensor network with size *n*, *p* denotes the probability that two neighboring nodes share common key or key information, which we call local connectivity. Let P_c be the probability that the graph is connected, which we call global connectivity. Erdös and Rényi [11] provided a theory how to determine p so that P_c is almost 1 (i.e. the graph is almost surely connected).

Erdös and Rényi [11] showed that, for monotone properties, there exists a value of p such that the property modes from "nonexistent" to "certain true" in a very large random graph *G(n,p)*. The function defining p is called the thresh hold function of property. Given a desired probability P_c for graph connectivity, the threshold p is defined by:

$$P_c = \lim_{n\to\infty} P_r[G(n,p)\ \ is\ \ connected] = e^{-e^{-c}} \tag{1}$$

where $p = \ln(n)/n + c/n$ and c is any real constant (2)

Eschenaeur and Gligor [1] analyzed that given n, they can find p and the expected degree of node (i.e. the average number of edges connecting that node with its graph neighbors) $d = p(n-1)$ for which the resulting graph is connected with desired probability P_c. We now prove that:

Theorem 1

After the *additional key-exchanging phase*, given *n* and the expected degree of node *d*, the probability P_c that the network graph is connected is always greater than that of the basic scheme (i.e. *key-exchanging* operation does not decrease the probability for graph connectivity).

Proof: Given *n* and *d*, we assume that P_c and P_c' are probabilities that the

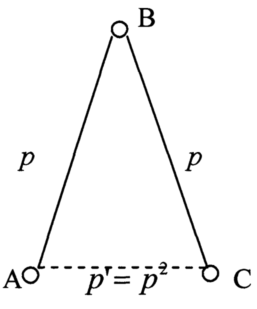

Fig. 2. The *Key-Exchanging* operation shifts secrete K_{AB} from B to C.

network graph is connected before and after *key-exchanging* operation, respectively. We now prove that $P_c' > P_c$.

Since *p* is the probability that a shared key exists between two nodes before *key-exchanging* phase, thus after *key-exchanging* phase this probability becomes $p' = p^2$. To prove this, we assume that there are three nodes A, B, and C (C is neighbor of A) as depicted in Fig.3. A and B share a common key with probability *p*. B and C share a common key with probability *p*. Thus, after the secret key K_{AB} is moved from B to C, the probability that a shared key exists between A and C is $p_{AC} = p' = p_{AB} \cdot p_{BC} = p^2$.

Since every node receives a secret key shared with each of its neighbors and vice versa, the probability that a shared key exists between two nodes in the network is:

$$p'' = 2p'(n-1) = 2p^2(n-1) \tag{3}$$

Because typically the number of sensor nodes $n > 1$ and $c > 0$, then from (2) and (3) we infer:

$$p = \frac{\ln(n)}{n} + \frac{c}{n} > \frac{1}{2n} \approx \frac{1}{2(n-1)}$$

$$\Leftrightarrow 2p(n-1) > 1$$

$$\Leftrightarrow 2p^2(n-1) > p$$

$$\Leftrightarrow p'' > p \tag{4}$$

Since P_c is directly proportional to *p* (or *d*), then from (4) we infer $P_c' > P_c$, i.e. *key-exchanging* operation does not decrease the probability of graph connectivity.

5. DISCUSSION

Obviously, shifting shared keys to each pair of neighboring nodes give a dramatic advantage for secure routing of DSNs. Every node can find the best path to its target through secure links. Since we do not reduce the number of key pre-distributed in entire sensor networks, but it may be increasing after key-exchanging phase, it is evident that the connectivity of the network is increasing. Consequently, other properties of the basic scheme are still maintained. One of the most advantages of this scheme is that it can be applied to whatever existing key pre-distribution schemes.

However, this scheme brings out many issues such that how to keep communication overhead of key-exchanging phase as minimum as possible. Another issue is how key-exchanging operation guarantees that the key is lost due to packet lost during transmission. We leave these issues for our future work.

6. CONCLUSION AND FUTURE WORK

We presented an improved scheme over Eschenaeur and Gligor scheme. This scheme gives an additional step, *key-exchanging phase*. By shifting the common keys to each pair of neighboring nodes, we can reduce significant computation and communication overhead of node-to-node communication while still guarantees original security of the basic scheme. This scheme, however, can be applied for all existing key pre-distribution schemes to improve the performance of secure routing for sensor networks.

In this paper, we have proposed a dramatic improvement over the basic scheme. In future work, we will investigate how much communication and computation overhead for *key-exchanging* operation. We also study how much our scheme supports to reduce energy consumption and computational cost for secure routing compared with the basic scheme.

References

[1] Laurent Eschenauer and Virgil D. Gligor. A key-management scheme for distributed sensor networks. In Proceedings of the 9th ACM Conference on Computer and Communication Security, pages 41–47, November 2002.

[2] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," IEEE INFOCOM, March 2004.

[3] W.Diffie and M.E.Hellman New direction in cryptograph. IEEE Transaction on Information Theory vol. 22 pp.644-654. November 1976

[4] B.C. Neumab sbd T. Tso. Kerberos: An authentication service for computer networks. IEEE communications Magazine. vol 40. no. 8. pp. 102-114. August 2002

[5] H.Chan, A. Perrig and D.Song. A random key predistribution schemes for sensor networks. in IEEE Symposium on Security and Privacy. Berkeley, California, May 11-14 2003 pp.197-213

[6] Haowen Chan, Adrian Perrig, and Dawn Song. Random key predistribution schemes for sensor networks. In IEEE Symposium on Security and Privacy, May 2003.

[7] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In ACM CCS 2003, pages 42–51, October 2003.

[8] Donggang Liu and Peng Ning. Establishing pairwise keys in distributed sensor networks. In ACM CCS 2003, pages 52–61, October 2003.

[9] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J. D. Tygar. SPINS: Security protocols for sensor networks. In Seventh Annual ACM International Conference on Mobile Computing and Networks (MobiCom 2001), July 2001.

[10] D. W. Carman, P. S. Kruus and B. J. Matt,"Constraints and Approaches for Distributed Sensor Network Security," dated September 1, 2000. NAI Labs Technical Report #00-010,

available at http://download.nai.com /products/media/nai/zip/nailabs-report-00-010-final.zip

[11] J. Spencer, The Strange Logic of Random Graphs, Algorithms and Combinatorics 22, Springer-Verlag 2000, ISBN 3-540-41654-4.

[12] F. Stajano, Security for Ubiquitous Computing, John Wiley and Sons, New York, Feb. 12, 2002, ISBN: 0-470-84493-0, 267 pp.

[13] Haowen Chan, A. Perrig, D. Song. Key Distribution Techniques for Sensor Networks. Springer-Verlag 2004, ISBN:1-4020-7883-8. pp. 277 – 303