An Efficient Key Pre-distribution Scheme for Secure Distributed Sensor Networks*

Sung Jin Choi and Hee Yong Youn**

School of Information and Communications Engineering,
Sungkyunkwan University, Suwon, Korea
{choisj, youn}@ece.skku.ac.kr

Abstract. Wide-spread deployment of sensor networks is emerging and it presents an economical solution to numerous problems. A number of applications are dependent on secure operation of the sensor network, however, and serious consequences are incurred if the network is compromised or disrupted. In the existing key pre-distribution scheme suitable for low power and resource sensor nodes, shared key is not guaranteed to be found and mutual authentication is not allowed. This paper thus proposes a new key pre-distribution scheme guaranteeing that any pair of nodes can find a common secret key between themselves by using the keys assigned by LU decomposition of a symmetric matrix of a pool of keys. Furthermore, it allows node-to-node mutual authentication. Analysis shows that the existing scheme requires a large number of keys in each sensor node to display a comparable performance as the proposed scheme. Therefore, the superiority of the proposed scheme is more substantial when the memory size of the sensor node is small.

Keywords: distributed sensor network, key pre-distribution, LU decomposition, mutual authentication, security.

1 Introduction

Wide-spread deployment of sensor networks is on the horizon. Networks of thousands of sensors may present an economical solution to some of the challenging problems: real-time traffic monitoring, monitoring of building safety (structural, fire, and physical security monitoring), military sensing and tracking, distributed measurement of seismic activity, real-time pollution monitoring, wild life monitoring, wild fire tracking, etc [1].

Distributed sensor networks (DSNs) share several characteristics with the traditional wireless networks. Both include arrays of sensor nodes that are battery powered, have limited computational capabilities and memory, and rely on intermittent wireless communication via radio frequency and, possibly, optical links. They also include data-collecting nodes which cache sensed data and make them available to the

* This research is supported by the Ubiquitous Autonomic Computing and Network Project, 21st Century Frontier R&D Program in Korea and the Brain Korea 21 Project in 2005.

** Corresponding author.

T. Enokido et al. (Eds.): EUC Workshops 2005, LNCS 3823, pp. 1088 – 1097, 2005.

application components of the network for processing, and control nodes which monitor the status of sensor nodes and broadcast simple commands to them. However, DSNs differ from the traditional wireless networks in several aspects, namely: their scale is a few orders of magnitude larger than that of wireless networks; they are dynamic in the sense that they allow addition and deletion of sensor nodes after deployment to extend the network or replace failing or unreliable nodes without physical contact; and they may be deployed in hostile areas where communication is monitored and the sensor nodes are subject to capture and manipulation by an adversary. These challenging operational requirements place equally challenging security constraints on the DSN design [2,3].

Many applications are dependent on secure operation of the sensor network, and have serious consequences if the network is compromised or disrupted. Also, when the sensor networks are deployed in a hostile environment, security becomes extremely important as they are prone to different types of malicious attacks. For example, an enemy can easily tap the information, imitate one of the sensor network nodes, or intentionally provide fault information to other nodes [4]. The problem here is how to secure the communication between the sensor nodes, i.e. how to set up secret keys between communicating nodes. Most earlier schemes use asymmetric cryptography to solve this problem [10]. However, theses schemes are often not suitable for distributed sensor network due to limited computation and energy power of the sensor nodes.

To address this issue a scheme has been recently proposed which is based on random key pre-distribution. However, it also has a shortcoming that a common key is not guaranteed to be found between two nodes wanting to communicate. This paper thus proposes a new key pre-distribution scheme which guarantees that any pair of nodes can find a secret key between themselves by using a pool of keys formed in the symmetric matrix format and the relevant property of LU decomposition of a matrix [13]. Furthermore, it allows node-to-node mutual authentication which the existing scheme does not support. Analysis shows that the existing scheme requires a large number of keys in each sensor node to display a comparable performance as the proposed scheme. Therefore, the superiority of the proposed scheme is more substantial when the memory size of the sensor node is small.

The rest of the paper is organized as follows. Section 2 discusses the existing key distribution approaches for sensor network, and Section 3 presents the proposed scheme. Section 4 analyzes and compares the performance of the proposed scheme with the earlier scheme, and finally concluding remark is given in Section 5.

2 Related Works

The traditional key exchange and distribution protocols based on the infrastructure of the internet using trusted third parties are impractical for large scale DSNs because of the network topology unknown prior to deployment, communication range limitation, intermittent sensor-node operation, and network dynamics, etc. To date, the only practical option for the distribution of keys to sensor nodes of large-scale DSNs whose physical topology is unknown prior to deployment would have to rely on key pre-distribution. Keys would have to be installed in the sensor nodes to accommodate secure connectivity between the nodes. However, the traditional key pre-distribution approach requires either a single mission key or a set of separate $n-1$ keys, each being privately shared with another node pair-wise, must be installed in every sensor node. This is an inadequate aspect for the DSNs [5].

There exist a number of key pre-distribution schemes. One solution is to let all the nodes carry a master secret key. Any pair of nodes can use this global master secret key to achieve key agreement and obtain a new pairwise key. This scheme does not exhibit desirable network resilience; if one node is compromised, the security of the entire sensor network will be compromised. Some existing studies suggest storing the master key in tamper-resistant hardware to reduce the risk [6], but this increases the cost and energy consumption of each sensor node. Furthermore, tamper-resistant hardware might not always be safe. Du et al. [7] proposed another key pre-distribution scheme which substantially improves the resilience of the network compared to other schemes. This scheme exhibits a threshold property; when the number of compromised nodes is smaller than the threshold, the probability that any node other than the compromised nodes is affected is close to zero. This desirable property lowers initial payoff of small scale network breaches to an adversary, and makes it necessary for the adversary to attack a significant portion of the network.

Blundo et al. [8] proposed several schemes which allow any group of some parties to compute a common key while being secure against collusion between some members of them. These schemes focus on saving communication cost while memory constraints are not placed on the group members. Perrig et al. [9] proposed SPINS, a security architecture specifically designed for sensor networks. In SPINS, each sensor node shares a secret key with the base station. Two sensor nodes cannot directly establish a secret key. However, they can use the base station as a trusted third party to set up a secret key.

Recently, Eschenauer and Gligor [10] proposed a random key pre-distribution scheme. Here a pool of random keys is selected from a key space. Each sensor node receives a subset of random keys from the pool before deployment. Any two nodes able to find one common key within their respective subsets can use it as their shared secret to initiate communication. Based on this scheme, Chan, Perrig, and Song [11] proposed a q-composite random key pre-distribution scheme, which increases the security of key setup such that an attacker has to compromise many more nodes to achieve a high probability of compromising communication. The difference between the q-composite scheme and the scheme in [10] is that q common keys ($q \geq 1$), instead of just a single one, are needed to establish secure communication between a pair of nodes. It was shown that network resilience against node capture is improved by increasing the value of q. The main issues in the random key pre-distribution approach are that a common key may not be found between a pair of nodes and node-to-node mutual authentication is not allowed. We next present the proposed scheme solving these problems.

3 The Proposed Scheme

In this section we present the basic features of the proposed scheme, deferring its analysis to the next section. First, we briefly describe how the proposed key pre-distribution scheme works. The proposed scheme uses a random graph like the Eschenauer's method [10]. It, however, guarantees that any pair of nodes can find a secret key between themselves along with mutual authentication.

3.1 Preliminaries

We capitalize some important properties of matrix in designing the key pre-distribution scheme.

Definition 1. If a square matrix K has the property $K^T = K$, where transpose of matrix K is denoted by K^T, we say that K is a symmetric matrix. A symmetric matrix means that $K_{ij} = K_{ji}$, where K_{ij} is the element in the ith row and jth column of matrix K.

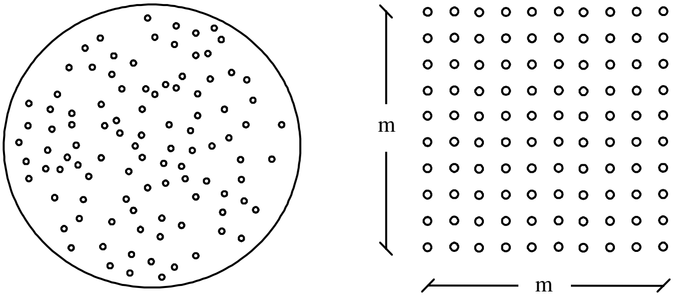
Definition 2. LU decomposition is to decompose an $m \times m$ matrix K into two matrices such that K = LU, where L is an $m \times m$ lower triangular and U is an $m \times m$ upper triangular matrix, respectively, i.e., product of the lower triangular matrix L and upper triangular matrix U gives rise to K.

3.2 The Proposed Key Distribution Scheme

We now explain the proposed key pre-distribution scheme. The key pre-distribution scheme consists of four off-line steps; namely generation of a large pool of keys (e.g., $2^{17} \sim 2^{20}$ keys), forming a symmetric matrix using the pool of keys, applying LU decomposition to the symmetric matrix, and key pre-distribution to each sensor node. We discuss the four steps next.

Step 1 (Generation of a large pool of keys (e.g., $2^{17} \sim 2^{20}$ keys)): We propose a key pre-distribution scheme using the random key approach. In the proposed key pre-distribution scheme each sensor node receives a subset of random keys from a large pool of keys before deployment. For communication, two nodes need to find one common key to use it as their shared secret key. Therefore, the base station first needs to generate a large pool of keys (e.g., $2^{17} \sim 2^{20}$ keys) in this step.

Step 2 (Forming a symmetric matrix using the pool of keys): Eschenauer's a random key pre-distribution scheme uses just a large pool of keys as shown in Figure 1(a). However, the proposed scheme uses a pool of keys formed in a symmetric matrix as shown in Figure 1(b).



(a) The pool of keys. (b) The pool of keys in symmetric matrix

Fig. 1. The pool of keys

Step 3 (Applying LU decomposition to the symmetric matrix): We apply LU decomposition to the symmetric matrix to let a pair of nodes always find a common key between themselves and raise the security by providing node-to-node mutual authentication.

Step 4 (Key pre-distribution): In this step every node is randomly assigned one row from the L matrix and one column from the U matrix, respectively. One and only one condition here is that the same row and column position are assigned such that L_{r_i}(*i*th row of L) and U_{c_i}(*i*th column of U) are assigned to each node.

(Finding a common key): Assume that node_x and node_y contains (L_{r_i} and U_{c_i}) and (L_{r_j} and U_{c_j}), respectively. When node_x and node_y need to find a common secret key between them, they first exchange their columns, and then compute a vector product as follows.

node_x: $L_{r\_i} \times U_{c\_j} = K_{ij}$
node_y: $L_{r\_j} \times U_{c\_i} = K_{ji}$

Recall that K is a symmetric matrix, and thus $K_{ij} = K_{ji}$. K_{ij} (or K_{ji}) is then used as a common key between node_x and node_y. Note that the proposed scheme allows any pair of nodes to always find a common secret key between themselves.

Example: We illustrate the proposed scheme using an example below.

Step 1: We first generate a large pool of keys using a random graph in this step, and assume here that we generate a pool of keys, S (−5~5).

Step 2: After we select (−2, 1, 2, 4) from the pool of keys, S, arrange them into a symmetric matrix K.

$$K = \begin{bmatrix} 2 & 4 & -2 \\ 4 & 1 & 2 \\ -2 & 2 & 1 \end{bmatrix} : \text{The pool of keys in a symmetric matrix}$$

Step 3: We apply LU decomposition to the symmetric matrix. We first calculate the elementary matrix $E_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$, and $E_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 6/7 & 1 \end{bmatrix}$ to derive L and U. Then we calculate $L = E_3E_2E_1A$ and $U = E_1^{-1}E_2^{-1}E_3^{-1}$. As a result, L and U are obtained as follows.

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -6/7 & 1 \end{bmatrix}, \ U = \begin{bmatrix} 2 & 4 & -2 \\ 0 & -7 & 6 \\ 0 & 0 & 29/7 \end{bmatrix}$$

Step 4: This step is for key pre-distribution, and assume that L_{r_3} and U_{c_3} are stored at node_x. Similarly, L_{r_2} and U_{c_2} are stored at node_y. When node_x and node_y need to find a secret key between them to securely communicate, they first exchange their columns, and then calculate the key value, respectively. Here the value turns out to be

2. Then they compare them for authentication. Since the values are same, they can authenticate each other and start communication using ‘2’ as the shared key. The process is summarized in Table 1.

Table 1. The operations for key authentication.

	node_x	node_y
After key pre-distribution	L_{r_3} (−1, −6/7, 1) U_{c_3} (−2, 6, 29/7)	L_{r_2} (2, 1, 0) U_{c_2} (4, −7, 0)
After column-exchange	L_{r_3} (−1, −6/7, 1) U_{c_2} (4, −7, 0)	L_{r_2} (2, 1, 0) U_{c_3} (−2, 6, 29/7)
After key-computation	$L_{r\_3} \times U_{c\_2} = 2$ (=K_{32})	$L_{r\_2} \times U_{c\_3} = 2$ (=K_{23})

3.3 Node-to-Node Mutual Authentication

The existing random key pre-distribution scheme does not allow node-to-node mutual authentication, but the proposed scheme based on the symmetric matrix of the keys does that as follows. Table 2 summarizes the process of node-to-node mutual authentication.

1. node_x sends U_{c_i} (the column it contains) to node_y .
   node_x → node_y : { U_{c_i} }

2. node_y: obtains K_{ji} by multiplying L_{r_j} with U_{c_i} received from node_x, and then sends U_{c_j} and K_{ji} to node_x.
   node_y : {$L_{r\_j} \times U_{c\_i} \rightarrow K_{ji}$}
   node_y → node_x : {U_{c_j}, K_{ji} }

3. node_x: obtains K_{ij} by multiplying L_{r_i} with U_{c_j} received from node_y, and compares it with K_{ji} received from node_y.
   node_x : {$L_{r\_i} \times U_{c\_j} \rightarrow K_{ij}$, check if $K_{ij} = K_{ji}$ }

4. If node_x verifies $K_{ij} = K_{ji}$, then sends K_{ij} to node_y.
   node_x → node_y : { K_{ij} }

5. node_y acknowledges K_{ij}: compares K_{ji} with K_{ij}.
   node_y : Check if $K_{ij} = K_{ji}$

Table 2. Node-to-node mutual authentication

Sensor node_x		Sensor node_y
L_{r_i}, U_{c_i}	$\xrightarrow{U_{c\_i}}$	L_{r_j}, U_{c_j} $L_{r\_j} \times U_{c\_i} \rightarrow K_{ji}$
$L_{r\_i\_} \times U_{c\_j} \bullet K_{ij}$	$\xleftarrow{U_{c\_j},\ K_{ji}}$	
$K_{ij} = K_{ji}$	$\xrightarrow{K_{ij}}$	$K_{ji} = K_{ij}$

4 Performance Analysis

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*. When *p* is zero, the graph does not have any edge, whereas when *p* is one, the graph is fully connected. Erdos and Renyi [12] showed that, for monotone properties, there exists a value of *p* such that the property moves from "nonexistent" to "certainly true" in a very large random graph. The function defining *p* is called the threshold function of the property. Given a desired probability P_c for graph connectivity, the threshold function *p* is defined by

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

Let *p* be the probability that a shared key exists between two sensor nodes, *n* be the number of nodes, and *d* be the expected degree as

$$d = p\times(n-1) = \frac{(n-1)(\ln(n)-\ln(-\ln(P_c)))}{n} \tag{2}$$

Figure 2 illustrates the plot of the expected degree of a node, *d*, as a function of the network size, *n*, for various values of P_c. The figure shows that the expected degree of a node needs to be increased by two to increase the probability that a random graph is connected by one order. Moreover, the curves of the plot are almost flat when *n* is large, indicating that size of the network has insignificant impact on the expected degree of a node required to have a connected graph.

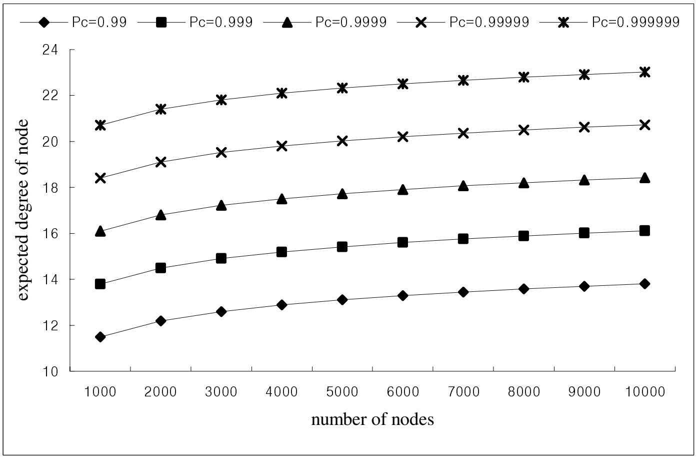


Fig. 2. Expected degree of a node for varying number of nodes

For a given density of sensor network deployment, let N be the expected number of neighbors within the communication range of a node. Using the expected node degree calculated above, the required local connectivity, $P_{required}$, can be estimated as follows [10].

$$\mathrm{P_{required}} = \frac{\mathrm{d}}{\mathrm{N}} = \frac{(\mathrm{n}-1)(\ln(\mathrm{n}) - \ln(-\ln(\mathrm{P_c})))}{\mathrm{nN}} \tag{3}$$

After we derive the required local connectivity, we decide the value *S* (the size of the key pool) and *k* (the number of keys in each node). The actual local connectivity is determined by these values. Note that *S* is not directly related to the sensor network, but *k* is related to the memory size of sensor node. Therefore, *k* needs to be as small as possible. We use P_{actual} to represent the actual local connectivity, which is the probability of any two neighboring nodes to find a common key between themselves. The link availability of any two nodes of the existing scheme [10] is then

$$1 - \text{Prob [a pair of nodes do not share a key].} \tag{4}$$

The probability that a pair of nodes, A and B, do not share a common key can be found using P_{actual}

$$P_{actual} = 1 - \frac{{}_{S}C_{k} \times {}_{S\text{-}k}C_{k}}{({}_{S}C_{k})^2} = 1 - \frac{((S-k)!)^2}{S!(S-2k)!}. \tag{5}$$

Since, *S* is very large, we use Stirling's formula for n!

$$n! \approx \sqrt{2\pi n}\left[\frac{n}{e}\right]^n. \tag{6}$$

To simplify the expression of P_{actual}, it is approximated as follows.

$$P_{actual} = 1 - \frac{(P-k)^{2P-2k+1}}{(P-2k)^{P-2k+\frac{1}{2}}}. \tag{7}$$

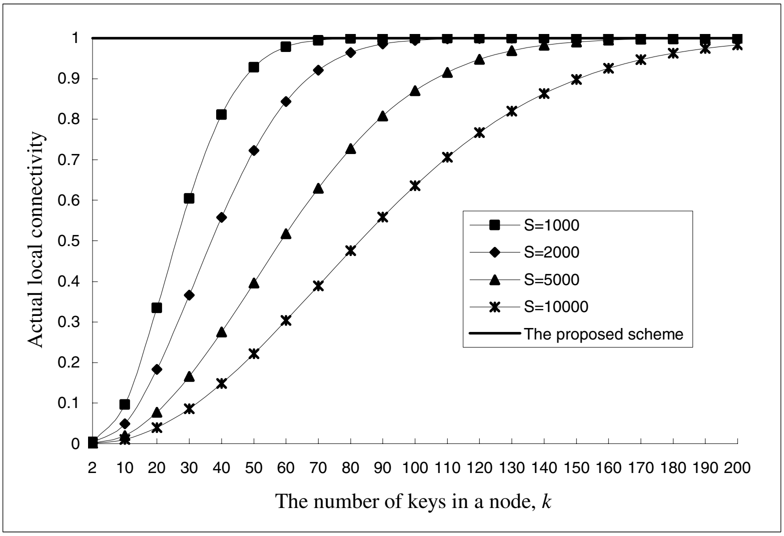


Fig. 3. Comparison of the connectivity of the proposed scheme with the existing scheme

Figure 3 compares the actual local connectivity of the proposed scheme with that of the existing scheme [10] when the size of the key varies from 2 to 200 for the size of the key pool *S* of 1000, 2000, 5000, and 10000. Observe from the figure that the local connectivity increases as the number of keys in a node increases for the existing scheme when the size of the pool of keys is fixed. The proposed scheme always allows the connectivity regardless of the number of keys per node. Note that, the superiority of the proposed scheme becomes more substantial when the memory size of the sensor node is small.

5 Conclusion and Future Works

Most earlier schemes proposed for security of distributed sensor network used asymmetric cryptography such as Deffie-Hellman key agreement or RSA. However, these schemes are often not suitable for distributed sensor network due to limited computation and energy resources of sensor node. The existing key pre-distribution scheme proposed to address this issue has a drawback of unguaranteed shared key between two nodes wanting to communicate. In this paper thus we have proposed a new key pre-distribution scheme guaranteeing that any pair of nodes can find a common key between themselves using the keys assigned by LU decomposition of a symmetric matrix. Also, it allows enhanced security by node-to-node mutual authentication. The existing scheme requires a large number of keys in each sensor node to display a comparable connectivity as the proposed scheme which allows 100% connectivity regardless of the number of keys. Therefore, the superiority of the proposed scheme is more substantial when the memory size of the sensor node is small. A new model considering not only connectivity but also security in a more formal way will be developed in the future.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci.: A survey on sensor networks: IEEE Communications Magazine, Vol. 40, no. 8, (2002) 102-114

[2] David W. Carman, Peter S. Kruus, and Brain J. Matt.: Constraints and approaches for distributed sensor network security: NAI Labs Technical Report #00-010, (2000)

[3] F. Stajano.: Security for Ubiquitous Computing: Jhon Wiley and Sons, ISBN 0-470-84493-0, (2002)

[4] J. Rabaey, J. Ammer, J. L. da Silva, D. Patel.: PicoRadio, Ad hoc wireless networking of ubiquitous low-energy sensor/monitor nodes, Workshop on VLSI, (2000)

[5] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varsheny.: A key management scheme for wireless sensor networks using deployment knowledge: Technical Report, Syracuse University, (2003)

[6] R. Anderson and M. Kuhn.: Tamper resistance – a cautionary note: Proceeding of the Second Usenix Workshop On Electronic Commerce, (1996) 1-11

[7] D. Liu and P. Ning.: Establishing pairwise keys in distributed sensor networks: Proceedings of the 10th ACM Conference on Computer and Communications Security, (2003) 52-61

[8] C. Blundo, A. D. Santix, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung.: Perfectly-secure key distribution for dynamic conferences: Lecture Note in Computer Science, Vol. 740, (1993) 471-486

[9] A. Perrig, R. Szewczyk, V. Wen, D. Cullar, and J. D. Tygar.: Spins (Security protocols for sensor networks): Proceeding of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking(MobiCom), (2001) 189-199

[10] L. Eschenauer and V. D. Gligor.: A key-management scheme for distributed sensor networks: Proceeding of the 9th ACM Conference on Computer and Communication security, (2002) 41-47

[11] H. Chan, A. Perrig, and D. Song.: Random key pre-distribution schemes for sensor networks: IEEE Symposium on Security and Privacy, (2003) 197-213

[12] Erdos and Renyi.: On random graphs I: Publ. Math. Debrecen, Vol 6, (1959) 290-297

[13] Sung Jin Choi, Hee Yong Youn.: A Novel Data Encryption and Distribution Approach for High Security and Availability Using LU Decomposition: The 2004 International Conference On Computational Science And Its Applications, LNCS 3046, (2004), 637-646