How to Break LU Matrix Based Key Predistribution Schemes for Wireless Sensor Networks

Bo Zhu*, Yanfei Zheng†, Yaowei Zhou† and Kefei Chen*
*Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, 200240 China
Email: zhubo03@gmail.com
†School of Information Security Engineering,
Shanghai Jiao Tong University, Shanghai, 200240 China
Email: yfzheng@sjtu.edu.cn

Abstract

A key predistribution scheme for Wireless Sensor Networks was proposed by Choi and Youn in 2005, which is based on LU decomposition of symmetric matrix. After that, several key predistribution schemes were designed on the basis of Choi and Youn's original scheme. In this paper, we carefully investigate a mathematical theorem about symmetric matrix, by following which adversaries could easily obtain the secret keys deployed via Choi and Youn's scheme. We also analyze all the schemes derived from Choi and Youn's and point out their vulnerabilities. In addition, we propose a revised scheme avoiding the security flaw.

1. Introduction

Wireless Sensor Networks (WSNs) are gaining popularity quickly because of their flexibility and low cost to solve a variety of real-world challenges [1]. The nodes in WSNs are equipped with many embedded devices, such as environmental sensors, signal processing chips, and wireless transceivers. The information collected by sensors can be processed by chips and transmitted between nodes. WSNs can be designed for numerous applications, such as remote locating, target tracking, and environmental monitoring in airports and battlefields. Since WSNs are always expected to deal with sensitive data and operate in critical environments, the mechanisms protecting sensor nodes and networks from potential security threats should be considered from the beginning of the system design [2].

When setting up a sensor network, one of the threshold requirements is to establish cryptographic keys for later use [3]. However, due to the characteristics of sensor nodes, their mechanisms to establish and manage cryptographic keys are significantly different from the ways of traditional networks. For instance, public-key cryptosystems, such as RSA and Diffie-Hellman, require a large amount of memory space and a significant computing ability, which are infeasible for the sensor nodes with limited capabilities. A practical key establishment approach is to assign cryptographic keys to sensor nodes before deploying them into the working environment, which is called *key predistribution scheme*. The most straightforward key predistribution scheme is to let the nodes in the network share a same key. The defect of this method is that had a single node been compromised by adversaries, the shared key will be known and the traffic of the whole network would be in danger. Another simple approach is to pre-configure a unique key for each pair of nodes, which would cost huge storage space and cannot scale well. In brief, how to construct a secure and efficient key predistribution scheme becomes one of the critical problems for researchers to solve.

In 2005, Choi and Youn proposed an LU matrix based key predistribution scheme (hereinafter, CY scheme) for WSNs [4]. In this scheme, cryptographic keys are first arranged into a symmetric matrix K, and then K is decomposed into the product of a lower triangular matrix L and an upper matrix U. The matrix L is used for reserved secret information of each node in WSNs, and U is used for public information exchange. CY scheme guarantees that any pair of nodes can establish a common key between them. After that, Park et al. proposed an efficient approach [5] to construct the matrices L and U in order to reduce the time overhead of CY scheme. Pathan and many other researchers proposed key predistribution schemes [6]–

978-1-4244-5113-5/09/$25.00

[8] by using the LU matrix based mechanism adopted from CY scheme. And later Choi and Youn proposed a multi-level key predistribution scheme [9] based on [4]. Wen et al. brought LU matrix based mechanism into hierarchical networks to improve system resilience and scalability [10].

1.1. Our Contributions

We find a mathematical theorem about the lower and the upper matrices from the decomposition of a symmetric matrix. In CY scheme, by following this theorem, adversaries can recover part of, even all of, the cryptographic keys deployed in the network if they obtain the secret information stored in a single node. This forces people to fully protect every node in the network, but it is infeasible to pay attention to every aspect of each node. Even worse, the security of the following LU matrix based key predistribution schemes all rely on the secure implementation of CY scheme. We carefully analyze all of LU matrix based schemes, and point out the influence of our discovery to each scheme. Finally, we propose a revised scheme in order to avoid the security flaw and provide more efficient ways to construct and transmit cryptographic information.

1.2. Organization of This Paper

The remainder of this paper is organized as follows. Section 2 describes and analyzes the original scheme proposed by Choi and Youn, and shows how to break this scheme by following a mathematical theorem. Section 3 briefly introduces all the following key predistribution schemes based on CY scheme, and shows the security influence to each scheme in the case that CY scheme is conquered. In Section 4, a revised symmetric matrix based key predistribution scheme is proposed. Section 5 gives the conclusion.

2. How to Break Choi and Youn's Scheme

This section first describes CY scheme to clarify the basic concepts involved in the series of LU matrix based schemes. Secondly, a mathematical theorem about symmetric matrix is presented and proved. Finally, a formal instruction and examples are given to illustrate how to break CY scheme in practical use.

2.1. Choi and Youn's Scheme

The definition of LU decomposition is given as follows.

Definition 1: LU decomposition is the process to decompose an $m \times m$ matrix K into two matrices L and U such that $K = LU$, where L is an $m \times m$ lower triangular matrix and U is an $m \times m$ upper triangular matrix.

The key predistribution process contains four steps:

Step 1: Generate a large pool of keys which are possible to be used for encrypting the communications between nodes.

Step 2: Construct a symmetric matrix K by randomly choosing elements from the key pool.

Step 3: Apply LU decomposition to K, and then get a lower triangular matrix L and an upper triangular matrix U.

Step 4: Assign keys to nodes. Every node should be randomly assigned one row of L and one column of U, where the row and the column have the same position in matrices, e.g., the i-th row of L (denoted by L_{r_i}) and the i-th column of U (denoted by U_{c_i}) should be assigned to a same node.

The key predistribution process should be done before the deployment of the whole WSNs. Each node in WSNs safely stores the keys assigned to it for later use, such as encryption and authentication.

For convenience, in the following context, the term "the i-th node" represents the sensor node storing the vectors L_{r_i} and U_{c_i}.

Establishing common keys. If the i-th node and the j-th node are going to establish a common cryptographic key, they first exchange the column vectors of U which they have respectively, i.e. U_{c_i} and U_{c_j}. In WSNs, the data will be broadcasted on the air. (One thing to keep in mind is that the information about L, e.g., L_{r_i} and L_{r_j}, will never be transmitted outside the nodes.) Then the two nodes compute the vector products as follows.

$$\text{the } i\text{-th node:} \quad k_{i,j} = L_{r\_i} U_{c\_j}$$
$$\text{the } j\text{-th node:} \quad k_{j,i} = L_{r\_j} U_{c\_i}$$

Because the matrix K is symmetric, $k_{i,j}$ is equal to $k_{j,i}$. The two nodes use the equivalent vector products as the common key to perform encryption, decryption, and authentication.

2.2. A Mathematical Theorem

One definition and two lemmas are given before proving the mathematical theorem.

Definition 2: Two vectors α and β are linearly dependently iff there exist two scalars c_1 and c_2, not both zero, such that $c_1\alpha + c_2\beta = 0$.

Lemma 1: Two vectors $\begin{bmatrix} a_1 & a_2 \end{bmatrix}$ and $\begin{bmatrix} b_1 & b_2 \end{bmatrix}$ are linearly dependent iff $a_1 b_2 = a_2 b_1$.

Proof:

(i) If $\begin{bmatrix} a_1 & a_2 \end{bmatrix}$ and $\begin{bmatrix} b_1 & b_2 \end{bmatrix}$ are linearly dependent, there exist two scalars c_1 and c_2, not both zero, such that

$$\begin{cases} c_1 a_1 + c_2 a_1 = 0 \\ c_1 a_2 + c_2 a_2 = 0 \end{cases}.$$

Without loss of generality, assuming $c_1 \neq 0$, we have

$$\begin{cases} a_1 = -\frac{c_2}{c_1} b_1 \\ a_2 = -\frac{c_2}{c_1} b_2 \end{cases},$$

which implies

$$\begin{cases} a_1 b_2 = -\frac{c_2}{c_1} b_1 b_2 \\ a_2 b_1 = -\frac{c_2}{c_1} b_2 b_1 \end{cases}.$$

Thus, $a_1 b_2 = a_2 b_1$.

(ii) Conversely, suppose that $a_1 b_2 = a_2 b_1$.

If $a_1 = b_1 = 0$, then the two vectors $\begin{bmatrix} 0 & a_2 \end{bmatrix}$ and $\begin{bmatrix} 0 & b_2 \end{bmatrix}$ are obviously linearly dependent.

If a_1 and b_1 are not both zero, then

$$\begin{aligned} & b_1 \begin{bmatrix} a_1 & a_2 \end{bmatrix} + (-a_1) \begin{bmatrix} b_1 & b_2 \end{bmatrix} \\ = \; & \begin{bmatrix} b_1 a_1 - a_1 b_1 & b_1 a_2 - a_1 b_2 \end{bmatrix} \\ = \; & \begin{bmatrix} 0 & 0 \end{bmatrix}. \end{aligned}$$

Thus, $\begin{bmatrix} a_1 & a_2 \end{bmatrix}$ and $\begin{bmatrix} b_1 & b_2 \end{bmatrix}$ are linearly dependent.

Combining (i) and (ii) completes the proof. □

Lemma 2: If $\begin{bmatrix} a_1 & a_i \end{bmatrix}$ and $\begin{bmatrix} b_1 & b_i \end{bmatrix}$ are linearly dependent for any i, where $1 \leq i \leq m$, and $a_1^2 + b_1^2 \neq 0$, then the two vectors $\begin{bmatrix} a_1 & a_2 & \cdots & a_m \end{bmatrix}$ and $\begin{bmatrix} b_1 & b_2 & \cdots & b_m \end{bmatrix}$ are linearly dependent.

Proof: Since $\begin{bmatrix} a_1 & a_i \end{bmatrix}$ and $\begin{bmatrix} b_1 & b_i \end{bmatrix}$ are linearly dependent, by following Lemma 1, we know $a_1 b_i = a_i b_1$ for any i, where $1 \leq i \leq m$.

The inequality $a_1^2 + b_1^2 \neq 0$ implies that a_1 and b_1 are not both zero, and then we have

$$\begin{aligned} & b_1 \begin{bmatrix} a_1 & a_2 & \cdots & a_m \end{bmatrix} + (-a_1) \begin{bmatrix} b_1 & b_2 & \cdots & b_m \end{bmatrix} \\ = \; & \begin{bmatrix} b_1 a_1 - a_1 b_1 & b_1 a_2 - a_1 b_2 & \cdots & b_1 a_m - a_1 b_m \end{bmatrix} \\ = \; & \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}. \end{aligned}$$

Thus, $\begin{bmatrix} a_1 & a_2 & \cdots & a_m \end{bmatrix}$ and $\begin{bmatrix} b_1 & b_2 & \cdots & b_m \end{bmatrix}$ are linearly dependent. □

Suppose L is an $m \times m$ lower triangular matrix, and U is an $m \times m$ upper triangular matrix. We use α_i to denote the i-th column vector of L (to differ from the row L_{r_i}), and β_i to denote the i-th row vector of U (to differ from the column U_{c_i}). The equation $K = LU$ can be expressed as

$$\begin{aligned} & \begin{bmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,m} \\ k_{2,1} & k_{2,2} & & \vdots \\ \vdots & & \ddots & \vdots \\ k_{m,1} & \cdots & \cdots & k_{m,m} \end{bmatrix} \\ = \; & \begin{bmatrix} l_{1,1} & & & \\ l_{2,1} & l_{2,2} & & \\ \vdots & & \ddots & \\ l_{m,1} & \cdots & \cdots & l_{m,m} \end{bmatrix} \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,m} \\ & u_{2,2} & & \vdots \\ & & \ddots & \vdots \\ & & & u_{m,m} \end{bmatrix} \\ = \; & \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_m \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix} \end{aligned}$$

We can easily get the following equations:

$$k_{i,j} = \begin{cases} l_{i,1} u_{1,j} + l_{i,2} u_{2,j} + \cdots + l_{i,i} u_{i,j} & (i < j) \\ l_{i,1} u_{1,j} + l_{i,2} u_{2,j} + \cdots + l_{i,i} u_{i,i} & (i = j) \\ l_{i,1} u_{1,j} + l_{i,2} u_{2,j} + \cdots + l_{i,j} u_{j,j} & (i > j) \end{cases}.$$

If the matrix product $K = LU$ is a symmetric matrix, then we have the following mathematical theorem.

Theorem 1: If there exists an s such that $l_{i,i}^2 + u_{i,i}^2 \neq 0$ for any i, where $1 \leq i \leq s \leq m$, then α_i^T and β_i are linearly dependent.

Proof:

(i) If $s = 1$, then $i = 1$.

For any j, where $1 \leq j \leq m$, we have

$$\begin{cases} k_{1,j} = l_{1,1} u_{1,j} \\ k_{j,1} = l_{j,1} u_{1,1} \end{cases}.$$

Since $k_{1,j} = k_{j,1}$, we have

$$l_{1,1} u_{1,j} = l_{j,1} u_{1,1}.$$

Deduced from Lemma 1, $\begin{bmatrix} l_{1,1} & l_{j,1} \end{bmatrix}$ and $\begin{bmatrix} u_{1,1} & u_{1,j} \end{bmatrix}$ are linearly dependent, where $1 \leq j \leq m$.

Furthermore, due to $l_{1,1}^2 + u_{1,1}^2 \neq 0$ and Lemma 2, we know that α_1^T and β_1 are linearly dependent.

(ii) If $2 \leq s \leq m$, then $1 \leq i \leq s$.

When $i = 1$, we can prove α_1^T and β_1 are linearly dependent just as the process in (i).

When $i = t \geq 2$, suppose α_r^T and β_r are linearly dependent for any r, where $1 \leq r \leq t-1$.

It is easy to see that $\begin{bmatrix} l_{t,r} & l_{j,r} \end{bmatrix}$ and $\begin{bmatrix} u_{r,t} & u_{r,j} \end{bmatrix}$, which are the parts of α_r^T and β_r, are linearly dependent, where $t \leq j \leq m$. This implies

$$l_{t,1} u_{1,j} = l_{j,1} u_{1,t}\ ,\ \cdots\ , l_{t,t-1} u_{t-1,j} = l_{j,t-1} u_{t-1,t}. \tag{1}$$

Since $k_{t,j} = k_{j,t}$, we can get the following equation

$$\begin{aligned} & l_{t,1} u_{1,j} + l_{t,2} u_{2,j} + \cdots + l_{t,t-1} u_{t-1,j} + l_{t,t} u_{t,j} \\ = \; & l_{j,1} u_{1,t} + l_{j,2} u_{2,t} + \cdots + l_{j,t-1} u_{t-1,t} + l_{j,t} u_{t,t}. \end{aligned} \tag{2}$$

Substituting (1) into (2) and simplifying give

$$l_{t,t}u_{t,j} = l_{j,t}u_{t,t}.$$

Similarly as (i), we can get that the two vectors $\begin{bmatrix} l_{t,t} & l_{t+1,t} & \cdots & l_{m,t} \end{bmatrix}$ and $\begin{bmatrix} u_{t,t} & u_{t,t+1} & \cdots & u_{t,m} \end{bmatrix}$ are linearly dependent, which implies α_t^T and β_t are linearly dependent.

Finally, from (i) and (ii), we know that α_i^T and β_i are linearly dependent for any i, where $1 \leq i \leq s$. □

Take the sample matrices in Choi and Youn's paper [4] to illustrate Theorem 1:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -6/7 & 1 \end{bmatrix} \text{ and } U = \begin{bmatrix} 2 & 4 & -2 \\ 0 & -7 & 6 \\ 0 & 0 & 29/7 \end{bmatrix}. \tag{3}$$

The two vectors $\alpha_1^T = \begin{bmatrix} 1 & 2 & -1 \end{bmatrix}$ and $\beta_1 = \begin{bmatrix} 2 & 4 & -2 \end{bmatrix}$ are linearly dependent, as well as $\alpha_2^T = \begin{bmatrix} 0 & 1 & -6/7 \end{bmatrix}$ and $\beta_2 = \begin{bmatrix} 0 & -7 & 6 \end{bmatrix}$.

Moreover, we can easily deduce the following corollary from Theorem 1.

Corollary 1: Given s, if $l_{t,t}^2 + u_{t,t}^2 \neq 0$ for any t, where $1 \leq t \leq s \leq m$, then

$$l_{i,s}u_{s,j} = l_{j,s}u_{s,i},$$

where $s \leq i \leq m$ and $s \leq j \leq m$.

2.3. Cryptanalysis of Choi and Youn's Scheme

One of the natural requirements when performing CY scheme is that the key matrix K should have full rank, which aims to minimize the relationship between different keys, otherwise some keys could possibly be determined by the others. If K has full rank, the lower and upper matrices L and U will also have full ranks. Under such circumstance, the elements on the main diagonal of L or U are all nonzero, and α_i^T and β_i are linearly dependent for any i, where $1 \leq i \leq m$. Even if K is rank deficient, as long as the elements with corresponding positions on the main diagonals of L and U are not both zero, the relationship between any pair of α_i^T and β_i still holds. Even if there are certain positions where $l_{i,i}$ and $u_{i,i}$ are both zero, the first many columns of L would still be linearly dependent with the corresponding rows of U.

Recall that when two nodes in WSNs want to build a secure connection, they should first exchange the column vectors U_{c_i} and U_{c_j} and establish a common key as $k_{i,j} = L_{r\_i}U_{c\_j} = L_{r\_j}U_{c\_i} = k_{j,i}$. The vectors U_{c_i} and U_{c_j} are transmitted in plaintext via wireless communication, so by eavesdropping adversaries can get the ratio between distinct column vectors of the upper triangular matrix U. By following Theorem 1, adversaries can also know the ratio between different row vectors of the lower triangular matrix L, which is the secret information preserved in the sensor nodes. Suppose that one of sensor nodes is compromised, which means adversaries obtain the cryptographic information in the compromised node, e.g., L_{r_i}, the adversaries could deduce the secret information stored in another node, e.g., L_{r_j}, after calculating the ratio between U_{c_i} and U_{c_j}.

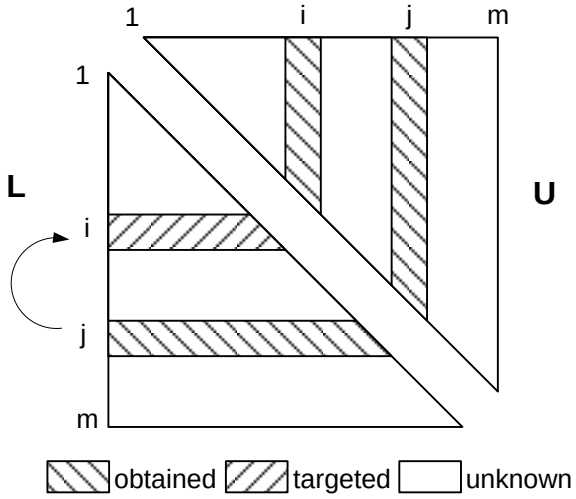

Figure 1. Calculating the i-th node's secret information by first compromising the j-th node, as the arrow shows.

The above analysis inspires us there is a formal approach for adversaries to figure out the secret information that they are interested in. The three steps to determine L_{r_i} are described as follows.

Step 1: Compromise one of the nodes, which contains L_{r_j}, where $i < j$. In this step, adversaries can choose the weakest nodes to break, such as the ones lacking enough hardware protection. Get L_{r_j} and U_{c_j}.

Step 2: Eavesdrop the communications of the targeted node, the i-th node. Get the information about U_{c_i}.

Step 3: Finally, if $u_{t,j} \neq 0$, by using Theorem 1 and the information about U_{c_i}, L_{r_j} and U_{c_j}, the vector L_{r_j} could be calculated as $l_{i,t} = \frac{u_{t,i}}{u_{t,j}} l_{j,t}$.

The whole process is shown in Figure 1.

Take the above sample matrices (3) to illustrate the attack steps. If the 3-rd node is compromised by adversaries, the vectors $L_{r\_3} = \begin{bmatrix} -1 & -6/7 & 1 \end{bmatrix}$ and $U_{c\_3} = \begin{bmatrix} -2 & 6 & 29/7 \end{bmatrix}^T$ are obtained. In this case, adversaries only need to eavesdrop and obtain U_{c_2}.

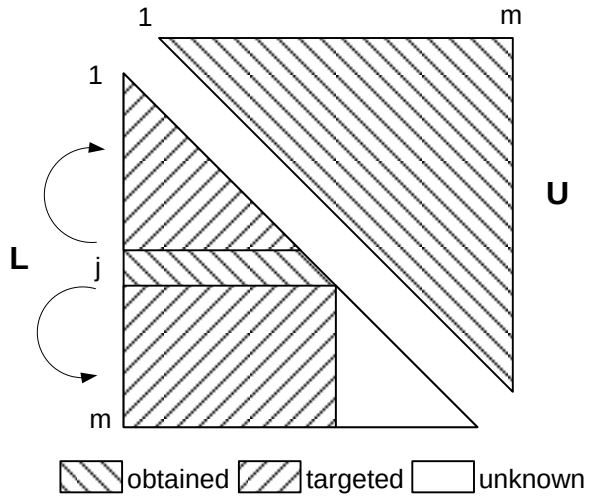

Figure 2. The coverage of the data which can be figured out if the j-th node is compromised.

As long as the 2-nd node is establishing new common keys, the U_{c_2} would be publicized on the air. After adversaries know $U_{c\_2} = \begin{bmatrix} 4 & -7 & 0 \end{bmatrix}^T$, they can easily calculate L_{r_2}, which is the secret information of the 2-nd node, just as follows.

$$\begin{cases} l_{2,1} = \frac{u_{1,2}}{u_{1,3}} l_{3,1} = \frac{4}{-2} \times (-1) & = 2 \\ l_{2,2} = \frac{u_{2,2}}{u_{2,3}} l_{3,2} = \frac{-7}{6} \times (-6/7) & = 1 \end{cases}$$

The result is $L_{r\_2} = \begin{bmatrix} 2 & 1 & 0 \end{bmatrix}$, which is the same as the original sample matrix.

More importantly, if adversaries want to get the secret information of one more node, they can simply repeat the above Step 2 and Step 3 without compromising another sensor node mentioned in Step 1. For example, if adversaries have known L_{r_3}, they can calculate both L_{r_2} and L_{r_1}. That is to say, if even a single sensor node is compromised, a large amount of the secret information of the other nodes can be figured out, and the entire network will be in danger. Figure 2 shows the data area of the matrix L can be figured out if adversaries first compromise the j-th node. More nonzero elements are contained in L_{r_j}, more data of the matrix L would be figured out. In the worst case, if adversaries compromise the m-th node, which carries the most nonzero information, they would be able to recover the entire matrix L. Especially, because of the publicity of the matrix U, adversaries can easily know which nodes contain the most information, which is a great convenience for adversaries to choose their first nodes to compromise.

The only exception is the case in which there exist zeros below the main diagonal of the matrix L. Take the matrices in Pathan's paper [6] for example:

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 \\ 4 & -1 & 3 & 1 \end{bmatrix} \text{ and } U = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & -3 \\ 0 & 0 & 0 & 2 \end{bmatrix}.$$

If it is only known that $L_{r\_3} = \begin{bmatrix} 3 & 0 & 1 & 0 \end{bmatrix}$, the vector L_{r_2} cannot completely be determined, because

$$l_{2,2} u_{2,3} = l_{3,2} u_{2,2}$$

that is

$$l_{2,2} \times 0 = 0 \times 1.$$

Under such circumstance, $l_{2,2}$ can be chosen arbitrarily. Thus $l_{4,2}$ must be obtained to calculate the rest of L_{r_2}. Therefore, if there exist zeros in the row vector contained in the compromised node, adversaries may not be able to compute all of the secret information which they are eager to know. However, such situation could possibly be avoided, because the elements with corresponding positions in L and U must be both zero or nonzero if K has full rank. Moreover, adversaries can compromise several nodes, and cooperatively use the nonzero parts of obtained vectors for calculation.

3. How to Break the other LU Matrix Based Schemes

This section briefly describes five distinct types of key predistribution schemes [5]–[10] based on CY scheme. The analysis shows that four of five types [5]–[7], [9], [10] are not secure in practical use, and Dai et al.'s scheme [8] is partially conquered.

3.1. Cryptanalysis of Park et al.'s Scheme

One of the shortcomings of LU matrix based schemes is that performing LU decomposition is rather time-consuming, especially when the dimension of the matrix K is very large. To reduce the time overhead of CY scheme, Park et al. proposed an efficient method to construct the matrices L and U: first form a lower triangular matrix L by randomly selecting elements from the key pool, and then calculate the upper triangular matrix U based on the predetermined matrix L. More details can be gotten from [5].

Park et al.'s scheme contains two major flaws. Firstly, although Park et al.'s method can significantly reduce the time overhead, their scheme still has the same problem as Choi and Youn's. Secondly, this scheme cannot ensure the elements of the matrix K are still in the key pool, even if the elements of matrix

L are selected from the key pool. It is entirely possible that some of the keys, the elements of $K = LU$, are weak keys [11] for certain symmetric algorithms, e.g., AES and IDEA. Such keys cannot be used to protect the communications. In sum, rather than improving the security of CY scheme, Park et al.'s works have served to bring one more uncertain problem.

3.2. Cryptanalysis of Pathan et al.'s Schemes

In 2006, two papers [6], [7] about LU matrix based key predistribution schemes were published by Pathan et al. One of the major contributions of Pathan et al.'s works is introducing an encoding mechanism to reduce the overhead of storage and communication. It is obvious to see that almost half of the elements of lower and upper triangular matrices are zero. Pathan et al. proposed that to store one row of L and one column of U in each sensor, it is only needed to store the nonzero elements and one value specifying the number of following zeros.

Nevertheless, just like the scheme proposed by Park et al., Pathan et al.'s schemes have contributed to improving the efficiency of CY scheme, but not better its security.

Moreover, an Intrusion Detection System (IDS) is mentioned in Pathan et al.'s papers [6], [7]. The IDS aims to detect the compromised sensor nodes, and revoke the corresponding vector information from the network. However, such an IDS cannot fight against the attack discussed in Section 2. Even if the compromised sensor nodes are removed or revoked from WSNs, the relationship between the matrices L and U, the radio of different vectors of the matrices, is not changed. The secret information stored in the single comprimsed node is enough for the starting point to break the whole network.

3.3. Cryptanalysis of Dai et al.'s Scheme

Dai et al.'s scheme [8] combines two key predistribution schemes, CY scheme and Blom's scheme. The scheme proposed by Blom [12] provides another kind of mechanism to predistribute shared keys. In Dai et al.'s scheme, when two nodes want to build a secure connection, they first compute two parts of the common key by using CY scheme and Blom's scheme separately, and then concatenate the two parts together as shown in Figure 3, or create the common key by using a hash function.

Since the implementation of CY scheme is not changed in Dai et al.'s works, Dai et al.'s scheme is still not safe due to the discussion in Section 2. The security

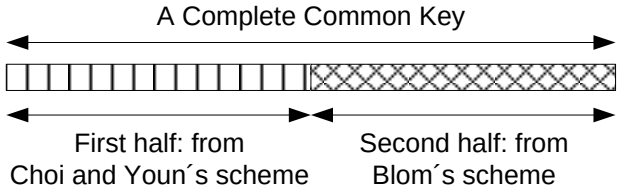

Figure 3. In Dai et al.'s scheme, a common key consists of two parts.

of Dai et al.'s scheme entirely relies on the secure implementation of Blom's scheme. Moreover, although Dai et al. claim their scheme has better performance and security than each of the two original schemes, it is inevitable to cost more storage and communication overhead to perform two distinct schemes.

3.4. Cryptanalysis of Choi and Youn's Multi-level Scheme

In 2007, Choi and Youn proposed a *multi-level* key predistribution scheme [9] based on their original scheme [4]. In this multi-level scheme, the upper triangular matrix U is further decomposed into the product of a diagonal matrix D and a new upper triangular matrix U', where the elements on the main diagonal of U' are all 1. Before being deployed into the network, each node is assigned the diagonal matrix D, and two vectors L_{r_i} and U'_{c_i}. When two nodes are to establish a common key, they exchange U'_{c_i} and U'_{c_j}, and compute the key as $L_{r\_i}DU'_{c\_j}$, which is equivalent to $L_{r\_j}DU'_{c\_i}$.

It is easy to see that the lower matrix L and the new upper matrix U' still satisfy the mathematical theorem described in Section 2, and U' has full rank. If a node in the network is compromised, the diagonal matrix D is also leaked to adversaries. Therefore, applying a further decomposition to the upper matrix U has no use to better the overall security level of the original scheme.

3.5. Cryptanalysis of Wen et al.'s Scheme

In Wen et al.'s scheme [10], a natural situation is supposed that wireless sensor nodes only have short transmission range, and thus it might be unnecessary to ensure every pair of nodes can establish a common key. WSNs are formed into a hierarchical architecture (see Figure 4) with a base station, cluster heads, and sensor nodes. The communications of sensor nodes are limited within the cluster where the nodes stay, and the data collected by sensor nodes is routed via

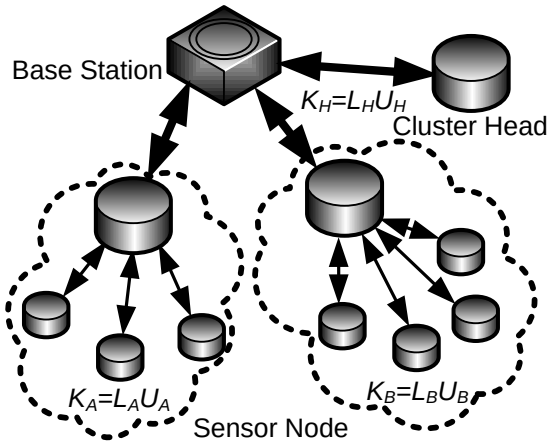

Figure 4. The hierarchical architecture of WSNs in Wen et al.'s scheme.

the cluster heads to the based station. In such a hierarchical network, Wen et al. proposed using several *small* CY schemes in clusters, rather than providing a *large* scheme which involves all of sensor nodes, cluster heads and base station. The cluster head and the sensor nodes in a cluster share a small symmetric key matrix, and the cluster heads and the base station share another small symmetric key matrix. This is significantly efficient when the network is large. For instance, a cluster head only needs to store four short vectors instead of two very long vectors.

If one node in a cluster is compromised, the secret information among the cluster will possibly be figured out. If a cluster head is compromised, the communications of all the cluster heads with the base station, and the transmitted data within the cluster where the compromised cluster head stays, would be decrypted by adversaries. Thus, to reduce the security influence, different clusters should use distinct key matrices, and cluster heads and the based station should be paid more attention to protect.

4. A Revised Key Predistribution Scheme

Although LU matrix based key predistribution schemes have certain security flaws, the thought of utilizing the properties of symmetric matrix decomposition does have some merits. To avoid the security flaws and reserve the advantages of these LU matrix based schemes, we propose a revised scheme here.

The key predistribution process of this revised scheme contains five steps:

Step 1: Generate a large pool which contains the keys that are possible to be used to secure the communications between nodes. There is no weak key [11] in the pool.

Step 2: Construct a symmetric matrix K, where the elements of K are randomly chosen from the key pool.

Step 3: Randomly construct an upper triangular matrix U no matter whether the elements are or are not in the key pool. And to reduce the overhead of storage and communication, the elements on the main diagonal of the matrix U are all set to a same nonzero number, e.g., 1.

Step 4: Multiply the matrix K and the inverse of the matrix U, and denote the result as M, i.e. $M = KU^{-1}$. Generally, M is not a lower triangular matrix. If M is a lower triangular matrix, we should go back to Step 3 to reconstruct U.

Step 5: Assign keys to nodes. Every node should be randomly assigned one row of M and one column of U, where the row and the column have the same position in matrices, e.g., the i-th row of M (denoted by M_{r_i}) and the i-th column of U (denoted by U_{c_i}) should be assigned to a same node.

The cryptographic information is assigned before sensor nodes are deployed into the network. When the i-th node and the j-th node are going *to establish a common key*, they first exchange the matrix U's column data which they have respectively, e.g., U_{c_i} and U_{c_j}. In our scheme, the two nodes only need to exchange the nonzero portion of the column vectors, and even do not need to send the element on the main diagonal (that is always equal to a constant number). After exchanging data, the two nodes extend the vectors respectively–append the omitted 0's and the constant number, and then compute vector products as follows.

$$\text{the } i\text{-th node:} \quad k_{i,j} = M_{r\_i} U_{c\_j}$$
$$\text{the } j\text{-th node:} \quad k_{j,i} = M_{r\_j} U_{c\_i}$$

The two nodes use the equivalent vector products $k_{i,j} = k_{j,i}$ as the common key to perform encryption, decryption, and authentication.

This revised key predistribution scheme has following advantages:

1) It does not have the security problem discussed in Section 2 and Section 3. This is because the proof of Theorem 1 is based on the special structures of the lower and upper triangular matrices. When the key matrix K is decomposed into the product of a general matrix M and a upper triangular matrix U, the property of Theorem 1 does not holds.

Table 1. Comparison of different schemes.

	Transmission Overhead	Storage Overhead	Predistribution Computation
CY scheme [4] and Wen's [10]	m	$2m$	LU decomposition
Park's [5]	m	$2m$	solving linear equations
Pathan's [6], [7]	$\frac{1}{2}m+\frac{1}{2}$	$m+2$	LU decomposition
Dai's [8]	$\frac{1}{2}m+\frac{1}{2}+$ extra Blom's	$m+2+$ extra Blom's	LU decomposition
CY multi-level scheme [9]	m	$3m$	solving linear equations
This paper	$\frac{1}{2}m-\frac{1}{2}$	$\frac{3}{2}m-\frac{1}{2}$	matrix multiplication

2) It avoids the high overhead of applying LU decomposition. It is much more efficient to compute the matrix product KU^{-1} than to decompose K. Because the elements on the main diagonal of the matrix U are all nonzero, U is always invertible.
3) A more efficient encoding mechanism is used in this scheme. The encoding mechanism is adopted from Pathan's works [6], [7]. However, in our scheme, the value specifying the number of following zeros is not needed because every node knows the length of the vector. Moreover, the unchanged elements on the main diagonal of U are not needed to be transmitted and stored. Such improvements can significantly reduce the overhead of the storage and communication, especially when the network consists of a huge amount of nodes.
4) This revised scheme still reserves most of the advantages of CY scheme, such as ensuring every pair of nodes can establish a common key.

The only one disadvantage realized so far is that the increased storage consumption in terms of storing the row vectors of a general matrix M instead of a lower triangular matrix L's. Table 1 shows the average transmission and storage consumptions of one node in distinct schemes, and their different approachs to compute keys during predistribution process. The parameter m is the dimension of the key matrix K.

5. Conclusion and Further Works

This paper presents and proves a mathematical theorem about symmetric matrix. The influence of this theorem to the security of LU matrix based key predistribution schemes has been fully analyzed, and the result shows that almost all of such schemes are not secure in practical use. If a single sensor node were compromised by adversaries, they would be able to recover all of the secret information of the whole network. We also propose a revised key predistribution scheme, reserving most of advantages of LU matrix based schemes, and outperforming in terms of the overhead of computing and transmission, but avoiding such security flaw. Our further works include comparing the performance of our proposed scheme with other key predistribution schemes for WSNs, such as Blom's scheme [12].

Acknowledgment

The authors would like to thank Dr. Mi Wen, Dr. Zheng Gong, and three anonymous referees for valuable comments on this paper.

This work is supported by National High-Tech Program (863) of China (under Grant No. 2006AA01Z422 and No. 2009AA01Z418) and National Basic Research Program (973) of China (under Grant No. 2007CB311201).

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, 2002.

[2] J. Walters, Z. Liang, W. Shi, and V. Chaudhary, *Security in Distributed, Grid, and Pervasive Computing*. Auerbach Publications, CRC Press, 2006, ch. 17.

[3] A. Perrig, J. Stankovic, and D. Wagner, "Security in Wireless Sensor Networks," *Commun. ACM*, vol. 47, no. 6, pp. 53–57, 2004.

[4] S. Choi and H. Youn, "An Efficient Key Pre-distribution Scheme for Secure Distributed Sensor Networks," in *Emerging Directions in Embedded and Ubiquitous Computing (EUC '05)*, ser. LNCS, vol. 3823. Springer, 2005, pp. 1088–1097.

[5] C. Park, S. Choi, and H. Youn, "A Noble Key Predistribution Scheme with LU Matrix for Secure Wireless Sensor Networks," in *Proc. Int. Conf. Computational Intelligence and Security (CIS '05) Part II*, ser. LNAI, vol. 3802. Springer, 2005, pp. 494–499.

[6] A. Pathan, T. Dai, and C. Hong, "A Key Management Scheme with Encoding and Improved Security for Wireless Sensor Networks," in *Proc. 3rd Int. Conf. Distributed Computing and Internet Technology (ICDCIT '06)*, ser. LNCS, vol. 4317. Springer, 2006, pp. 102–115.

[7] A. Pathan, T. Dai, and C. Hong, "An Efficient LU Decomposition-based Key Pre-distribution Scheme for Ensuring Security in Wireless Sensor Networks," in *Proc. 6th IEEE Int. Conf. on Computer and Information Technology (CIT '06)*, Sept. 2006, pp. 227–227.

[8] T. Dai, A. Pathan, and C. Hong, "A Resource-Optimal Key Pre-distribution Scheme with Enhanced Security for Wireless Sensor Networks," in *Proc. 9th Asia-Pacific Network Operations and Management Symposium (APNOMS '06)*, ser. LNCS, vol. 4238. Springer, 2006, pp. 546–549.

[9] S. Choi and H. Youn, "MKPS: A Multi-level Key Pre-distribution Scheme for Secure Wireless Sensor Networks," in *Proc. 12th Int. Conf. Human-Computer Interaction (HCII '07) Part II*, ser. LNCS, vol. 4551. Springer, 2007, pp. 808–817.

[10] M. Wen, Y. Zheng, H. Li, and K. Chen, "A Hierarchical Composition of LU Matrix-Based Key Distribution Scheme for Sensor Networks," in *Proc. 11th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'07)*, ser. LNAI, vol. 4819. Springer, 2007, pp. 608–620.

[11] J. Daemen, R. Govaerts, and J. Vandewalle, "Weak Keys for IDEA," in *Advances in Cryptology – CRYPTO '93*, ser. LNCS, vol. 773. Springer, 1994, pp. 224–231.

[12] R. Blom, "An Optimal Class of Symmetric Key Generation Systems," in *Proc. EUROCRYPT 84*. Springer, 1985, pp. 335–338.