Efficient Pairwise Key Establishment Scheme for Sensor Networks

Wang Lei
Department of Computer and Information Science
Fujian University of Technology
Fuzhou City, Fujian Province, 350014
P.R.China
http://www.fjit.edu.cn/

Abstract: - Security communication is an important requirement in many sensor network applications. A new kind of pairwise key establishment scheme based on hierarchical hypercube model \(H(k,u,m,v,n)\) for clusters-deployed sensor networks distribution model is presented. Theoretic analysis and experimental figures show that the new pairwise key model has better performance and provides higher possibilities for sensor to establish pairwise key, compared with previous related works.

Key-Words: - Sensor networks; Security; Pairwise key; Hypercube; Algorithm; Model

1 Introduction

Security schemes of pairwise key establishment, which enable sensors to communicate with each other securely, play a fundamental role in research on security issue in wireless sensor networks. As one of the most fundamental security services, pairwise key establishment enables the sensor nodes to communicate securely with each other using cryptographic techniques.

Eschenauer and Gligor proposed a basic probabilistic key pre-distribution scheme for pairwise key establishment [1]. In the scheme, each sensor node randomly picks a set of keys from a key pool before the deployment so that any two of the sensor nodes have a certain probability to share at least one common key. Chan et al. further extended this idea and presented two key pre-distribution schemes: a \(q\)-composite key pre-distribution scheme and a random pairwise keys scheme. The \(q\)-composite scheme requires any two sensors share at least \(q\) pre-distributed keys. The random scheme randomly picks pair of sensors and assigns each pair a unique random key [2]. Inspired by the studies above and the polynomial-based key pre-distribution protocol [3], Liu et al. further developed the idea addressed in the previous works and proposed a general framework of polynomial pool-based key pre-distribution [4]. A similar approach to those schemes described by Liu et al was independently developed by Du et a. [5]. Rather than on Blundo's scheme their approach is based on Blom's scheme [6]. In 2004, Liu proposed a new hypercube-based pairwise key pre-distribution scheme [7], the analysis shows that, when perfect security against node compromise is required, the hypercube-based scheme can support a larger network by adding more dimensions instead of increasing the storage overhead on sensor nodes. Though hypercube-based scheme has many attractive properties, it requires any two nodes in sensor networks can communicate directly with each other. This strong assumption is impractical in most of the actual applications of the sensor networks.

In this paper, we present a kind of new cluster-based distribution model of sensor networks, and for which, we propose a new pairwise key pre-distribution scheme. The main contributions of this paper are as follows: Combining the deployment knowledge of sensor networks and the polynomial pool-based key pre-distribution, we setup a cluster-based topology that is practical with the real deployment of sensor networks. Based on the topology, we propose a novel cluster distribution based hierarchical hypercube model to establish the pairwise key. The key contribution is that our scheme does not require the assumption of all nodes can directly communicate with each other as the previous schemes do, and it still maintains high probability of key establishment, low memory overhead and good security performance.

2 Model of Clusters-Deployed Sensor Networks

In some actual applications of sensor networks, sensors can be deployed through airplanes. Supposing that the deployment rounds of sensors are \(k\), and the communication radius of any sensors
is $r$, then the sensors deployed in the same round can be regarded as belonging to a same Cluster. We assign a unique cluster number $l$ ($1 \leq l \leq k$) for each cluster. Supposing that the sensors form a connected graph in any cluster after deployment through airplanes, and then the Fig.1 presents an actual model of clusters deployed sensor networks.

![Fig.1 An actual model of clusters deployed sensor networks](image)

From Fig.1, it is easy to know that, for a given node $A$, there exist lots of nodes in the same cluster of $A$, which can be communicated directly with $A$, since the nodes are deployed densely in a cluster. But there exist much less nodes in a cluster neighboring to the cluster of $A$, which can be communicated directly with $A$ since the two clusters are not deployed at the same time.

### 3 Hierarchical Hypercube Model

**Definition 1 (Key Predistribution):** The procedure, which is used to encode the corresponding encryption and decryption algorithms in sensor nodes before distribution, is called Key Predistribution.

**Definition 2 (Pairwise Key):** For any two nodes $A$ and $B$, if they have a common key $E$, then the key $E$ is called a pairwise key between them.

**Definition 3 (Key Path):** For any two nodes $A_0$ and $A_k$, when there has not a pairwise key between them, if there exists a path $A_0, A_1, A_2, \ldots, A_{i-1}, A_i$ and there exists at least one pairwise key between the nodes $A_i$ and $A_j$ for $0 \leq j \leq k-1$ and $1 \leq j \leq k$, then the path consisted of $A_0, A_1, A_2, \ldots, A_{i-1}, A_i$ is called a Key Path between $A_0$ and $A_k$.

**Definition 4 (n-dimensional Hypercube):** An $n$-dimensional Hypercube (or $n$-cube) $H(v,n)$ is a topology with the following properties: (1) It is consisted of $v^n - 1$ edges, (2) Each node can be coded as a string with $n$ positions such as $b_1b_2\ldots b_n$, where $0 \leq b_1, b_2, \ldots, b_n \leq v-1$, (3) Any two nodes are coded neighbors, which means that there is an edge between them, iff there is just one position different between their node codes.

**Definition 5 (k-levels Hierarchical Hypercube):** Let there are $N$ nodes totally, then a $k$-levels Hierarchical Hypercube named $H(k,u,m,v,n)$ can be constructed as follows:

1) The $N$ nodes are divided into $k$ clusters averagely, and the $[N/k]$ nodes in any cluster are connected into an $n$-dimensional Hypercube: In the $n$-dimensional Hypercube, any node is encoded as $i_1i_2\ldots i_n$, which are called In-Cluster-Hypercube-Node-Codes, where $0 \leq i_1, i_2, \ldots, i_n \leq v-1$. $v=\lceil\sqrt[k]{N} \rceil$ equals to an integer not less than $i$. So we can obtain $k$ such kind of different hypercubes.

2) The $k$ different hypercubes obtained above are encoded as $j_1j_2\ldots j_m$, which are called Out-Cluster-Hypercube-Node-Codes, where $0 \leq j_1, j_2, \ldots, j_m \leq u-1$, $u=[\sqrt[k]{N}]$. And the nodes in the $k$ different hypercubes are connected into $m$-dimensional hypercubes according to the following rules: The nodes with same In-Cluster-Hypercube-Node-Codes and differentOut-Cluster-Hypercube-Node-Codes are connected into a $m$-dimensional hypercube.

The graph constructed through above steps is called a $k$-levels Hierarchical Hypercube abbreviated as $H(k,u,m,v,n)$, in which any node $A$ can be encoded as $(i,j)$, where $i=(i_1i_2\ldots i_n)$, $0 \leq i_1, i_2, \ldots, i_n \leq v-1$ is the In-Cluster-Hypercube-Node-Code of node $A$, and $j=(j_1j_2\ldots j_m)$, $0 \leq j_1, j_2, \ldots, j_m \leq u-1$ is the Out-Cluster-Hypercube-Node-Code of node $A$.

### 4 Mapping Clusters Deployed Sensor Networks to $H(k,u,m,v,n)$ Model

Obviously, from the description in section 3 and 4, we can know that the clusters deployed sensor network can be mapped into a $k$-levels- hierarchical hypercube model as follows:

At first, the $k$ clusters in the sensor network can be mapped into $k$ different levels (or hypercubes) in the $k$-levels- hierarchical hypercube model. Then, the sensor nodes in each cluster can be encoded with the In-Cluster-Hypercube-Node-Codes, and the sensor nodes in the $k$ different clusters with the same In-Cluster-Hypercube-Node-Codes can be encoded with the Out-Cluster-Hypercube-Node-Codes according to the definition 5 respectively.

Consequently, the whole sensor network has been mapped into a $k$-levels- hierarchical hypercube model.

### 5 $H(k,u,m,v,n)$-based Pairwise Key Predistribution Algorithm

In order to overcome the drawbacks of polynomial-based and polynomial pool-based key
predistribution algorithms, this paper proposed an innovative $H(k,u,m,v,n)$ model-based key predistribution scheme and pairwise key establishment algorithm, which combines the advantages of polynomial-based and key pool-based encryption schemes, and is based on the KDC and polynomials pool-based key predistribution models.

The new $H(k,u,m,v,n)$ model-based pairwise key establishment algorithm includes three main steps: (1) Generation of the polynomials pool and key predistribution, (2) Direct pairwise key discovery, (3) Path key discovery.

### 5.1 Generation of the Polynomials Pool and Key Predistribution

Supposing that, the sensor network includes $N$ nodes, and is deployed through $k$ different rounds. Then we can predistribute keys for each sensor node on the basis of the $H(k,u,m,v,n)$ model as follows:

Step 1: Key setup server randomly generates a bivariate polynomials pool $P = \{ f_i^{l}, f_j^{l} \}$ $(x,y)$, where $0 \leq i \leq 2$, $0 \leq j \leq 2$, $1 \leq l \leq n$, and assigns a unique polynomial ID to each bivariate polynomials over a finite field $\mathbb{F}_q$, then assigns a unique polynomial ID to each bivariate polynomial in $P$.

Step 2: In each round, key setup server assigns a unique node ID: $(i,l,j_1,j_2,...,j_m)$ to each sensor node from small to big, where $0 \leq i \leq 2$, $0 \leq j_1,j_2,...,j_m \leq n-1$.

Step 3: key setup server assigns a unique cluster ID: $l$ to all the sensor nodes deployed in the same round, where $1 \leq l \leq k$.

Step 4: key setup server predistributes $m+n$ bivariate polynomials $\{ f_i^{l}, f_j^{l} \}$ $(i,l,j_1,j_2,...,j_m)$ and the corresponding polynomial IDs to the sensor node deployed in the $l$th round and with ID $(i,l,j_1,j_2,...,j_m)$.

### 5.2 Direct Pairwise Key Discovery

If the node $A(i_1,i_2,...,i_n,j_1,j_2,...,j_m)$ in the sensor network wants to establish pairwise key with a node $B(i'_{1},i'_{2},...,i'_{n},j'_{2},...,j'_{m})$, then node $A$ can establish pairwise key with the node $B$ through the following methods.

Firstly, node $A$ computes out the distance between itself and node $B$: $d = d_1 + d_2$, where $d_1 = d_2(i_1,i_2,...,i_n)$ and $d_2 = d_2(i_1,j_2,...,j_m,j'_{2},...,j'_{m})$. If $d = 1$, then node $A$ obtains the direct pairwise key between itself and node $B$ according to the following theorem 1:

**Theorem 1:** For any two sensor nodes $A(i_1,i_2,...,i_n,j_1,j_2,...,j_m)$ and $B(i'_{1},i'_{2},...,i'_{n},j'_{1},j'_{2},...,j'_{m})$ in the sensor network, supposing that the distance between nodes $A$ and $B$ is $d = d_1 + d_2$, where $d_1 = d_2(i_1,i_2,...,i_n)$ and $d_2 = d_2(i_1,j_2,...,j_m,j'_{2},...,j'_{m})$. If $d = 1$, then there exists a direct pairwise key between nodes $A$ and $B$.

According to theorem 1, we present the detailed description of the direct pairwise key discovery algorithm as follows:

**Step 1:** Obtain the node IDs and cluster IDs of the source node $A$ and destination node $B$.

**Step 2:** Compute out the distance between nodes $A$ and $B$: $d = d_1 + d_2$.

**Step 3:** If $d_1 = 1$, $d_2 = 0$, then select out a common polynomial share of nodes $A$ and $B$ from $\{ f_1^{l}, f_2^{l},...,f_{n}^{l} \}$ to establish direct pairwise key.

**Step 4:** If $d_1 = 0$, $d_2 = 1$, then select out a common polynomial share of nodes $A$ and $B$ from $\{ f_1^{l}, f_2^{l},...,f_{n}^{l} \}$ to establish direct pairwise key.

**Step 5:** Otherwise, there exists no direct pairwise key between nodes $A$ and $B$. And then turn to the following path key discovery process.

### 5.3 Path Key Discovery

If $d = 1$, then node $A$ can establish path key with node $B$ according to the following theorem 2:

**Theorem 2:** For any two sensor nodes $A(i_1,i_2,...,i_n,j_1,j_2,...,j_m)$ and $B(i'_{1},i'_{2},...,i'_{n},j'_{1},j'_{2},...,j'_{m})$ in the sensor network, supposing that the distance between nodes $A$ and $B$ is $d = d_1 + d_2$, where $d_1 = d_2(i_1,i_2,...,i_n)$ and $d_2 = d_2(i_1,j_2,...,j_m,j'_{2},...,j'_{m})$. If $d = 1$, then there exists a path key between nodes $A$ and $B$.

According to theorem 2, we present the detailed description of the path key discovery algorithm as follows:

**Step 1:** Compute out the intermediate nodes $(i_{1},i'_{1},i_{2},...,i_{n},j_{1},j_{2},...,j_{m})$, $(i'_{1},i'_{2},...,i'_{n},j'_{1},j'_{2},...,j'_{m})$, and $(i'_{1},i'_{2},...,i'_{n},j'_{1},j'_{2},...,j'_{m},j_{1},j_{2},...,j_{m})$, $(i'_{1},i'_{2},...,i'_{n},j'_{1},j'_{2},...,j'_{m},j_{1},j_{2},...,j_{m},j_{1},j_{2},...,j_{m})$ from the source node $A(i_1,i_2,...,i_n,j_1,j_2,...,j_m)$ and the destination node $B(i'_{1},i'_{2},...,i'_{n},j'_{1},j'_{2},...,j'_{m})$.

**Step 2:** In those nodes series $A(i_1,i_2,...,i_n,j_1,j_2,...,j_m)$, $(i'_{1},i'_{2},...,i'_{n},j'_{1},j'_{2},...,j'_{m})$, $(i_{1},i_{2},...,i_{n},j_{1},j_{2},...,j_{m})$, $(i'_{1},i'_{2},...,i'_{n},j'_{1},j'_{2},...,j'_{m},j_{1},j_{2},...,j_{m})$, $(i'_{1},i'_{2},...,i'_{n},j'_{1},j'_{2},...,j'_{m},j_{1},j_{2},...,j_{m})$, $(i'_{1},i'_{2},...,i'_{n},j'_{1},j'_{2},...,j'_{m},j_{1},j_{2},...,j_{m})$, $B(i'_{1},i'_{2},...,i'_{n},j'_{1},j'_{2},...,j'_{m})$, the
neighboring nodes select their common polynomial share to establish direct pairwise key.

From theorem 2, it is easy to know that any source node \( A \) can compute out a key path \( P \) to the destination node \( B \) according to the above algorithm, when there are no compromised nodes in the sensor network. Once the key path \( P \) is computed out, then node \( A \) can send messages to \( B \) along the path \( P \) to establish indirect pairwise key with node \( B \). Fig.2 presents a example of key path establishment.

![Fig.2 Key Path establishment example](image)

**Fig.2 Key Path establishment example**

For example: In the above Fig.2, node \( A((012),(1234)) \) can establish pairwise key with node \( B((121),(2334)) \) through the following key path: \( A((012),(1234)) \rightarrow C((112),(1234)) \rightarrow D((122),(1234)) \rightarrow E((121),(1234)) \rightarrow F((121),(2234)) \rightarrow B((121),(2334)) \), where node \( F \) shall route through nodes \( G, H, I, J \) to establish direct pairwise key with node \( B \).

According to the properties of \( H(k,u,m,v,n) \) model, we can prove that the following theorem by combing the proof procedure of theorem 2:

**Theorem 3:** Supposing that there exist no compromised nodes in the sensor network, and the distance between node \( A \) and \( B \), then there exists a shortest key path with \( k \) distance between node \( A \) and \( B \) logically. That is to say, node \( A \) can establish indirect pairwise key with node \( B \) through \( t-1 \) intermediate nodes.

In actual applications, since the sensors are deployed by airplanes, so the \( H(k,u,m,v,n) \) model formed at key predistribution stage will be destroyed after deployment. So the two nodes neighboring to each other in the logical space \( H(k,u,m,v,n) \) are perhaps not neighboring in the actually deployed sensor network space, then they cannot communicate with each other directly. Therefore, in the actual deployed sensor network environment, any two nodes \( A \) and \( B \) may need more than \( t-1 \) intermediate nodes to establish path pairwise key with each other.

5.4 Dynamic Path Key Discovery

The path key discovery algorithm proposed in the above section can establish a key path correctly, when there are no compromised nodes in the whole sensor network only, since the key path is computed out beforehand. And the proposed algorithm cannot find an alternative key path when there exist some compromised nodes or some intermediate nodes not in the communication radius, even that there exists other alternative key paths in the sensor network. From the following example we can know that there are many parallel paths in the \( H(k,u,m,v,n) \) model for any two given source and destination nodes, since the \( H(k,u,m,v,n) \) model is high fault-tolerant[8,9].

![Fig.3 Alternative key path establishment example](image)

**Fig.3 Alternative key path establishment example**

For example: Considering the key path establishment example given in the above section based on Fig.2: \( A((012),(1234)) \rightarrow C((112),(1234)) \rightarrow D((122),(1234)) \rightarrow E((121),(1234)) \rightarrow F((121),(2234)) \rightarrow B((121),(2334)) \), supposing that node \( F((121),(2234)) \) has compromised, then from Fig.3, we can know that there exists another alternative key path as \( A((012),(1234)) \rightarrow C((112),(1234)) \rightarrow D((122),(1234)) \rightarrow E((121),(1234)) \rightarrow M((121),(1334)) \rightarrow B((121),(2334)) \), which can be used to establish the indirect pairwise key between node \( A \) and \( B \), where node \( E \) shall route through nodes \( D \) and \( K \) to establish direct pairwise key with node \( M \), and node \( M \) shall route through nodes \( N, O, G, H, I, J \) to establish direct pairwise key with node \( B \).

Since the sensors are source limited, so they are easy to die or out of the communication radius, therefore the algorithm proposed in the above section cannot guarantee to establish correct key path efficiently. In this section, we will propose a dynamic path key discovery algorithm as follows, which can improve the probability of key path effectively:

**Input:** Sub-sensor network \( H(k,u,m,v,n) \), which has some compromised /fault sensors and fault links, And two reachable nodes \( A(a_1…a_m) \) and \( B(a'_{1}…a'_{m}) \)
and $B(b_1...b_nb'_1...b'_m)$ in $H(k,u,m,v,n)$, where $a'_i \neq b'_i, i \in [1..s], a'_i=b'_i, t \geq s$.

Output: A correct key path from node $A$ to $B$ in $H(k,u,m,v,n)$.

Step 1: Obtain the code strings of node $A$ and $B$: 
$A \leftarrow (a_1...a_u,a'_1...a'_m)$, $B \leftarrow (b_1...b_u,b'_1...b'_m)$, where $a_i, b_j \in [0,u-1], a'_i, b'_j \in [0,v-1]$.

Step 2: If $a'_1...a'_m = b'_1...b'_m$, then node $A$ can find a route to $B$ according to the routing algorithms of hypercube [10].

Step 3: Otherwise, node $A$ can find a route to $C(b_1...b_u,a'_1...a'_m)$. Then let $I_0=C(b_1...b_u,a'_1...a'_m)$, $I_1=b_1...b_u,a'_2...a'_m)$, ..., $I_k=B(b_1...b_u,b'_1...b'_m)$, and each node $I_i$ in the above nodes series find a route to its neighboring node $I_{i+1}$ on the basis of the location information (Detailed routing algorithms based on location information can see the references[11-12]).

Step 4: Algorithm exits. If such kind of a correct key path exists, then through which node $A$ can establish an indirect pairwise key with node $B$. Otherwise, node $A$ fails to establish an indirect pairwise key with node $B$. And node $A$ will tries again to establish an indirect pairwise key with node $B$ some time later.

6 Algorithm Analyses

According to the former description and analyses, it is easy to know that the above newly proposed algorithm has the following properties:

Property 1: When there exist no fault and compromised nodes, by using new pairwise key predistribution scheme based on $H(k,u,m,v,n)$ model, the probability of direct pairwise key establishment between any two nodes can be estimated as $P=(m(u-1)+n(v-1))(N-1)$, where $N$ is the total number of nodes in the sensor network, and $N=\#_n \cdot v^2$.

Proof: Since the predistributed pairwise keys for any node $F_A$ =\{$f^1_{i_1,i_2,...,i_u}\rightarrow (i_1,y)\}, \ldots, f^n_{i_1,i_2,...,i_u}\rightarrow (i_n,y); f^1_{j_1,j_2,...,j_n} \rightarrow (j_1,y)\}, \ldots, f^m_{j_1,j_2,...,j_n} \rightarrow (j_n,y)$\} in the newly proposed algorithm. Obviously, in the logical hypercube formed by the nodes in the same cluster of node $A$, there are $n(v-1)$ nodes, which have direct pairwise key with node $A$. And in the logical hypercube formed by the nodes in different clusters from that of node $A$, there are $m(u-1)$ nodes, which have direct pairwise key with node $A$. Therefore, there are totally $m(u-1)+n(v-1)$ nodes, which have direct pairwise key with node $A$. So, the probability of pairwise key establishment between any two nodes can be estimated as $P=(m(u-1)+n(v-1))(N-1)$, since the whole sensor network has $N$ sensor nodes in all. [End Proof]

Fig.4 comparison between the probability of direct pairwise key establishment for any two nodes and the dimension $n$, when the sensor network has different total nodes, and use the new pairwise key predistribution scheme based on $H(8,2,3,v,n)$ model

From Fig.4, it is easy to know that by using new pairwise key predistribution scheme based on $H(k,u,m,v,n)$ model, the probability of direct pairwise key establishment between any two nodes decreases with the increasing of the scale of the sensor networks, and in addition, the probability of direct pairwise key establishment between any two nodes decreases with the increasing of the dimension $n$, when the scale of the sensor network is fixed.

Theorem 4: Supposing that the total sensors is $N$ in the sensor network, then when $u \geq v^2$, the probability of direct pairwise key establishment between any two nodes, when using the key distribution scheme based on the hypercube model $H(v,p)$, is smaller than that when using the key distribution scheme based on the $H(k,u,m,v,n)$ model.

As for the conclusion of theorem 4, we give an example to illustrate.

Supposing that the total number of nodes in the sensor network is $N=2^{14}$, and $H(k,u,m,v,n)=H(16,4,2,2,10)$, $H(v,p)=H(10,14)$, then the probability of direct pairwise key establishment between any two nodes based on the $H(k,u,m,v,n)$ model is $P=(m(u-1)+n(v-1))(N-1) = (24-1)+10(2-1)/(2^{14}-1) = 16/(2^{14}-1)$, but the probability of direct pairwise key establishment between any two nodes based on the $H(v,p)$ model is $P=(n(v-1))(N-1) = (14(2-1))/(2^{14}-1) = 14/(2^{14}-1)$.

Supposing that the total number of nodes in the sensor network is $N$. Fig.5 illustrates the comparison between the probability of direct pairwise key establishment between any two nodes based on the $H(k,u,m,v,n)$ model and the probability of direct
pairwise key establishment between any two nodes based on the $H(v,p)$ model, when $u=4$ and $v=2$.

Fig.5 comparison between the probability of direct pairwise key establishment and scale of sensor networks for $H(v,n)$ and $H(k,u,m,v,n)$ models

From Fig.5, it is easy to know that by using new pairwise key predistribution scheme based on $H(k,u,m,v,n)$ model, the probability of direct pairwise key establishment between any two nodes decreases with the increasing of the scale of the sensor networks, and in addition, the probability of direct pairwise key establishment between any two nodes decreases with the increasing of the dimension $n$, when the scale of the sensor network is fixed.

7 Conclusion

An innovative pairwise key predistribution scheme based on hierarchical hypercube is designed, by combing the good properties of the Polynomial Key and Key Pool encryption schemes, in which nodes are not needed to be able to communicate with each other directly such as that the hypercube model-based algorithm shall need. So, the traditional pairwise key predistribution scheme based on hypercube model is only a special case of the new algorithm proposed in this paper. Theoretical and experimental analyses show that the newly proposed scheme is an efficient pairwise key establishment scheme that is suitable for the cluster deployed sensor networks. How to build group keys for sensor networks by using the proposed hierarchical hypercube model is our next research works.

Acknowledgements: The authors’ research supported in part by Programme of Young Scientists and Technicians of Fujian Province with Grant 2005J051.

References: