International Journal of Security and Its Applications
Vol. 10, No. 7 (2016) pp.119-134
http://dx.doi.org/10.14257/ijsia.2016.10.7.11

A Key Management and Cross-layer Routing Scheme for Wireless Sensor Networks

Fengjun Shang Xiaolin Deng Yongkui Zhou

College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;
Author to whom correspondence should be addressed; E-mail: shangfj@cqupt.edu.cn

Abstract

In this paper, a key management and security multi-hop routing scheme based on cross layer (SMRCL) was proposed for wireless sensor networks. The core of SMRCL has two parts as following. Firstly, a key management based on polynomial and time deployment for wireless sensor networks is proposed. In this scheme, the sensor nodes are divided into several deployment groups and deployed the network according to the time. Moreover polynomial is used instead of common key, which the redundant polynomial is deleted when a deployment group has been deployed to the network. Secondly, a multi-hop routing algorithm is proposed based on cross-layer, which can be used in large scale. In the algorithm, we can use the information of the MAC layer, consider the cross-layer design by the ratio of the packet sending/reception rate and queue length to improve the network load ability. Simulation results demonstrate that SMRCL can save energy and balance the network load and improve the network lifetime. Comparison of the probability of the sensor connectivity, performance of resistance to the node capture and the sensor node overheads, show that the scheme can guarantee the high probability of node secure connectivity, improves the performance of resistance to the node capture, reduces the node storage overhead effectively.

Keywords*: Wireless sensor network, Key management, Cross layer, polynomial, Multi-hop*

1. Introduction

In recent years, with the progress of sensor, computer and communication technology, Wireless Sensor Networks (WSNs) has been rapid development and attracted wide attention throughout the world. Wireless sensor network consists of many wireless sensor nodes by the methods of multi-hop and self-organization of wireless communication [1, 2].The same as the traditional network, wireless sensor network also exists the problem of network security. However, compared with traditional network, wireless sensor network usually deploys on unsupervised or rugged condition, so it lacks of stable network infrastructure as a guarantee. The deployment of Sensor Network has the characteristics of wide areas, high density, and strong randomness of positions. Therefore, the topological structure of network is not predicted before deployment. And, the network is often applied in enemy's area in the application of military. The network is easily attacked and encounters security threats, because the characteristics of opening distribution and wireless broadcast communication of the network have potential safety hazard, and it restricts energy, bandwidth, processing and storage capacity. Due to the restriction of structure and properties of wireless sensor network nodes and randomness adversity of nodes' deployment condition, wireless sensor network is faced with many security challenges. Current network mechanism could not be totally adapt to this area because the specialty of wireless sensor network compared with traditional network. The security

ISSN: 1738-9976 IJSIA

problems have become the main challenge for the development and application of wireless sensor network.

Due to the limitation of nodes' resources, the lack of stable foundation about network, and easily damaged of nodes etc., the scheme and protocol for key management usually can't be applied to wireless sensor network directly. Therefore, key management is always the emphasis in the academia [3]. For the past few years, the researches on key management in wireless sensor network have made great progress. The schemes and protocols may use different technologies, and the points are also different. According to the various cryptosystem of network, the scheme of key management in wireless sensor network could be divided into symmetric keys management and asymmetric keys management. In the asymmetric keys management scheme, sensors' nodes that transmit and receive data encrypt and decrypt with one key. Because of low demand for sensors' nodes, symmetric key is suitable for Wireless sensor network. Therefore, symmetric keys management is the mainstream in this field [4]. However, in asymmetric keys management, there are different keys to encrypt and decrypt between two parties. Asymmetric keys are more security than symmetric keys, while it costs more because of its complicated encryption algorithm. So, someone thought asymmetric keys was not suitable for wireless sensor network. But, with further research, some asymmetric cryptographic algorithms have been optimized to be applied to wireless sensor network. O. Feng Tao put forward a scheme of group key based on ECC [5]. Then, O. Arazi also proposed DoS mitigation mechanism which based on RSA [6]. In addition, WEI Jiang-Hong proposes a two-party attribute based authenticated key exchange protocol with provable security in the standard model [7]. And the fast authentication and key protocol was extended by Kotzanikolaou to be applied to heterogeneous network, which improve the scalability of networks [8]. Yu wangke put forward a scheme of keys management in the way of bilinear pairings encryption based on elliptic curve [9].

According to whether the key ring is determined before deployment, the key management could be divided into random key management and fixed key management. In random key management, nodes select some keys randomly as own key ring from key pools or key spaces. Although the way of distribution of keys is simple and there is probability between nodes when they connect each other, chosen randomly caused some useless keys that waste storage space of nodes. Fixed key management determined key ring of node through geographic information [10], symmetric polynomial [4] or symmetric BIBD [11] and so on. Under the circumstance of low cost of nodes' storage, we ensure the probability of safe connectivity of nodes to 1. But, it usually costs much more in calculate and storage. Besides, there is restriction to deploy nodes.

According to different network topology, key management could be divided into distributed key management and hierarchical key management. In distributed key management, sensor nodes have same resources, energy and status. It generates the key agreement and update between nodes in the way of pre-distribution and mutual cooperation, so its distributivity is better. In hierarchical key management, sensor nodes are split into two types: common sensor nodes and cluster head, whose status and functions are different. Key management manages mainly through cluster head, so it's lower demand for common one. Cluster head might have powerful resources and energy. Therefore, its trapping has greater threat on network.

In this paper, a key management and multi-hop routing scheme based on cross layer was proposed for wireless sensor networks. In Section 2, a key management based on polynomial and time deployment is introduced, which the sensor nodes are divided in to several deployment groups and deployed to the network according to the time, polynomial is used instead of common key, redundant polynomial is deleted when a deployment group has been deployed to the network. In Section 3, a multi-hop routing algorithm is proposed based on cross-layer, which it can use the information of the MAC layer by the ratio of the packet sending/reception and queue length to improve the network load

ability. In Section 4, simulations and analysis are given. In Section 5, the conclusions are obtained.

2. Network Model and Group Designing

2.1. Network Model

In this paper, the scheme of key management is homogeneous, static, preallocated and distributed. Sensor nodes referred to have the same functions and resources. Also, the nodes' resources, such as calculation, communication and storage, are limited. Besides, base station that deployed in monitoring area has enough resources and energy to make direct communication among any nodes in the network, so that there is no multi-hops transmission through another node. Sensor nodes at a distance, however, need middle nodes to transmit when they send data to the base station. While the network is running, it is necessary to deploy some new nodes in network. The reasons are that nodes may die for exhaustion, nodes are captured and extend networks' function *etc*.

In order to describe the scheme better, we define some parameters below:

BS: base station.

SN: sensor node.

m: the number of keys selected from key pool S_i by sensor nodes.

w: the number of foreign key pools in sensor nodes.

r: the times that sensor nodes are deployed by time in network.

E_i: Article *i* deployment in network $1 \le i \le r$.

S_i: Article *i* key pool $1 \le i \le r+w$.

G_i: Article *i* deployment group in network which is sensor nodes' set deployed in E_i.

S: the size of the key pool which is the number of key in the key pool.

KID_{ij}: the KeyID of keys in the key pool S_i is unique.

2.2. Group Designing

We deploy sensor nodes in network via grouping method, which includes the deploy model based on region partition and the other one by time [12, 15].In this paper, we mainly discussed the model based on time [15]. On the basis of the key management according to time, we introduce bivariate symmetric polynomial instead of common keys to make internet against capture better. Besides, in the course of sharing polynomial, we encrypt the identification of polynomial with one-way hash key in order to prevent nodes from targeted attack via the compromising polynomial identification. Group designing includes sensor nodes, bivariate symmetric polynomial and the generation of one-way key chain:

(1) We deploy nodes in form of group in time order, and divided all of sensor nodes into r times for deployment. Everytime we deploy the nodes, the nodes in this group belong to the same group, denoted U_i, obviously, $1 \le i \le r$.

(2) Having completed to divide sensor nodes, we design the group for key pool while generate *r+w* key pools that the size of r is same as the number of deployment group. Key pool consists of multiple bivariate symmetric polynomials.

Definition 1: $f(x, y)$ is bivariate symmetric polynomial with t derivative over $GF(q)$:

$$f(x,y)=\sum_{i,j=0}^{l} a_{ij}x^{i}y^{j} = a_{00}+a_{01}x^{0}y^{1}+\ldots+a_{ll}x^{l}y^{l} \quad (1)$$

where q is the big primes adapted to the key length, a_{ij} belong to finite fields $GF(q)$. Bivariate symmetric polynomials have two important property:

(1) $f(x,y)$ is symmetric, that is $f(x,y)=f(y,x)$;

(2) l-order polynomial has $l-1$-order security. When the quantity of captured nodes with same polynomial is not more than $l-1$, it is unable to calculate the coefficient a_{ij} of polynomial, so we can't crack it.

The specific implementation process of polynomial group designing: firstly, base station generate a biggish key space made up of multiple bivariate symmetric polynomials $f(x,y)$; secondly, with no reposition, we draw $r+w$ time constinuously, which every time draws bivariate symmetric polynomials from key space, to constitute $r+w$ key pools respectively $S_1,S_2,\ldots,S_{w+r}$. Certainly, there are s keys in every key pool. And one polynomial has a unique KeyID KID_{ij}. Where, i represents the serial number of polynomial's key pool $1\le i\le r+w$, j represents the serial number of keys in key pools $1\le j\le S$.

According to group number r, we compute one-way key chain whose length is r+1 with one-way hash function, such as the formula (2).

Definition 2: One-way Key Chain:

$$Chain = K_0 \xleftarrow{hash} K_1 \xleftarrow{hash} \ldots K_{r-1} \xleftarrow{hash} K_r \quad (2)$$

in the formula, $K_{i-1}=hash(K_i)$, where K_r is initial key. In the chain, even if K_{i-1} *has been compromised, attacker can't get* K_i *via* K_{i-1}.

2.3. Key Management Scheme Based on Polynomial and Time Deployment

According to node grouping design, all the sensor nodes are divided and deployed by r times in the light of deployment group U_i which called as node deployment event, called E_i. The E_i contains three stages: initialization, discovering shared polynomial and deleting redundant polynomial.

2.3.1. Initialization: Initialization begin before deployment of deployment group. Firstly, before the first time to deploy node, base station divides nodes into r deployment group according to the method of group designing, respectively $U_1,U_2,\ldots,U_r$. Also, it distributes a unique nodes' identification SN_{ij} for every sensor node, i represents which deployment group belong to, j represents the node's identification in its deployment group. Beyond that, the nodes in the same deployment group only have one i.

Secondly, on the basis of the method of key grouping, there are $r+w$ key pools that contain various of polynomials' keys f(x,y), respectively $S_1,S_2,\ldots,S_{r+w}$. All nodes in deployment group U_i select m polynomials randomly from key pool S_i as a part of key ring, that S_i is called home key pool of U_i. Furthermore, all nodes of U_i select w key pools from key pools S_{i+1}, S_{i+2}, …, S_{w+r} randomly as foreign key pools. Then, electing m polynomials from every foreign key pool randomly, there are $m(1+w)$ polynomials totally to make up key ring of sensor nodes in U_i. $f(x,y)$ stores in nodes as the form of $f(ID_i,y)$, where ID_i represents the ID of sensor nodes which contain polynomials.

Actually every node store $m(1+w)$ unary l-order polynomials. In deployment group U_i, the home key pools and the foreign one are all named associated key pool of U_i.

Finally, sensor nodes keep one-way hash function and one-way key K_{i-1} in U_i. K_{i-1} is the i^{th} key of one-way key chain.

2.3.2. Discovering Shared Polynomial: Discovering Shared Polynomial stage starts after U_i deployed at wireless sensor network. During this stage, every sensor nodes search polynomial shared with neighbor node. This stage completes via the identification that sensor nodes sign polynomial in the range of single hop. While sensor nodes are broadcast identification, attacker could get all polynomial identifications through eavesdropping attack. These information make attacker capture target nodes easily, which cause biggest threat to the network with minimal captured nodes. In order to prevent sensor nodes from identification compromise, we use one-way hash key K that was deployed in nodes in advance to encrypt polynomial identification E (KID_{ij},K).The detail shows as below:

Firstly, sensor nodes SN_{ij} broadcast its identification and all the keys' identification in key ring encrypted with one-way hash key to its neighbor in range of single hop.

Secondly, sensor nodes SN_{pq} should jude whether it shared the same key pool with SN_{ij} when it receives broadcast data from SN_{ij}. If they share key pool, nodes will search shared polynomial. When nodes find the identification of shared polynomial, SN_{pq} is going to generate $f(ID_{pq},ID_{ij})$ with the node's identification ID_{ij} that is taken in shared polynomial $f(ID_{pq}, y)$.

Finally, the keys which S_{pq} and SN_{ij} shared is operated Exclusive OR. Then, the result is taken into one-way hash function to generate the communication key between two sensor nodes. $K_c = Hash(k_1 \oplus k_2 \oplus k_i \oplus ... \oplus k_l), 1 \le l \le m$, w=2 is the size of sensor nodes' key ring. The communication key K_C is created through the above process. And network deployment event E_i is completed.

2.3.3. Deleting Redundant Polynomial: As can be seen from the above, the duration of event E_i just is the time T_{int} of process of searching shared key. Because the process of discovering shared key only occur between neighbor nodes, so we consider the time of discovering keys is also short, that means T_{int} is a small value. After E_i completed, nodes' key rings contain two parts of polynomial keys in deployment group $U_j(j \le i)$.A part of key pools numbers i, the other part numbers after i. We define that S_i is the home key pool of U_i, while U_i is being deployed. Besides, we need to select w foreign key pools. And these pools are ordered after S_i, that is the identification of foreign pools is greater than the one of home ones. In that way, after E_i finished, the polynomial keys in home key pool S_i of U_i is no longer the shared keys between deployment groups. It means the polynomial keys selected randomly from home key pool S_i become redundant polynomial keys. After U_i complete to discover shared polynomial keys, in other word, timing from U_i deploying after T_{int} time, we could delete redundant keys from sensor nodes' key ring. When all deployment group complete to deploy, mean after E_r finish, the keys of all sensor nodes will be deleted.

First of all, at the initialization of network, in addition to extracting keys randomly from associated key pool, BS also distribute the key one-way K_{i-1} of chain to the nodes of group U_i, such as loading K_0 of one-way key chain in U_1 beforehand.

Then, at the period of discovering shared key, BS broadcast a message ADV_i to network, which means begin to deploy. BS encrypt ADV_i with the key K_i of one-way key chain, in order to prevent attacker to counterfeit it and cause that sensor nodes delete the wrong key which hasn't been redundant. Meanwhile, the broadcast message includes K_i. And the message of beginning deployment is referred to as:

$$\{K_i,E(ADV_i,K_i),mac(E(ADV_i,K_i),K_i)\}$$

Finally, sensor nodes, in network, start time and verify whether hash(K_i) is equal to K_{i-1} stored in sensor node after it received ADV_i. If they are equal, then it shows that ADV_i comes from BS. If it succeeds in proving, sensor nodes decrypt to acquire ADV_i with K_i and test MAC certification. Timing after T_{int} that means after discovering shared key, sensor nodes delete redundant polynomial key in key ring and K_{i-1}.It takes very short time which is T_{del} to delete redundant polynomial key.

The process above just is the step to delete redundant polynomial key. It has following three advantages:

(1) It is able to reduce the cost of memory of sensor nodes. All the polynomials stored in nodes will be deleted after all groups are deployed in network, so that there is no memory cost for polynomial keys;

(2) It reduces the number of polynomial keys which attackers capture from sensor nodes;

(3) When the redundant polynomial keys are deleted, sensor nodes only broadcast the keys' identification of the existing key rings during discovering shared keys. In that way, it cuts down the cost of communication.

3. Multi-hop Routing based on Cross-layer

In [13], it proposed a cross-layer routing protocol, which is only used the ratio of the packet sending rate and receiving rate as a standard to judge the node when compute node ratio. This trend value can only show that the node may be busy and unable to determine whether the node is busy. We can use the information of the MAC layer, consider the cross-layer design, rich nodes busy judgment and improve the network load ability. In [14], a novel routing protocol based on integrated metrics is proposed for Infrastructure/Backbone wireless mesh networks combining optimized link state routing protocol with cross-layer design theory.

In our multi-hop routing based on cross-layer, MAC layer and network layer determine which node to select. The MAC layer provides two parameters: the ratio of the packet sending rate and packet reception rate and queue length. Packet reception rate represents the amount of data received by the node per unit time and the packet sending rate represents the amount of data sent by the node per unit time. When the packet sending rate is greater than the packet reception rate, it is to say that the ratio between the two is greater than 1 indicates that the current node is sending more data and receiving less data, so the buffer is increasing and there is not busy possibility on this node. When the packet sending rate is less than the packet reception rate, it is to say that the ratio between the two is less than 1 indicates that the current node is sending less data and receiving more data, so the buffer exists a decreasing trend and there is a busy trend. As the previous analysis, the separate use of the data packet reception rate and packet sending rate ratio can't be good to determine whether the node is in busy state. The ratio between the two is just a trend, and we also need to consider the actual state of the current node. So at the same time it uses the queue length to represent the actual size of the current node's buffer. The queue length represents the packet length stored in node buffer at some time point. The longer the queue, the more unhandled packet is; the shorter the queue, the less unhandled packet is. If the node's data reception rate is more than the data sending rate, the longer the queue length, we can determine the node is busy. In other cases we can say the node in non-busy state. Formulate above description is $V = {V_s}/{V_a}$, $V_s = {S_s}/{T}$,

$V_a = S_a / T$ and $T = t - t'$, where V_S indicates the node's packet sending rate at some time; V_a indicates the node's receiving packet receiving rate for a moment. When V<1, says that the node packet receiving rate is greater than the packet sending rate, congestion may occur; when V>1, says that the node packet sending rate is greater than the packet receiving rate, congestion will not occur. S_s indicates the number of packets sending in a certain time period. S_a indicates the number of packets receiving in a certain time period. T indicates sampling time discrepancy.

Set threshold value L_b, when L is more than L_b, we can assume that the node can accommodate very limited data. And at this time if V <1, we can be sure that the node is busy. Threshold value L_b is experience which is not given directly by the formula. It is different in different network environments. Value of L_b will affect the parameters such as delay, throughput and packet loss rate. The average delay changes if we only consider the case of the average delay in the network and L_b ranges between 43 and 50. Therefore, we use 46 as the threshold.

4. Simulations and Analysis

4.1. The Secure Connectivity Probability

According to the description of discovering shared polynomials, to establish communication keys between neighbor nodes and communicate safely, there should share a polynomial key at least. It should meet two conditions to share polynomial keys between neighbor nodes:

Conditions 1: The two neighbor nodes have the same key pool in the deployment group, that is shared key pool;

Conditions 2: The two sensor nodes have a same polynomial key at least which is one of the keys extracted from shared key pool.

According to condition 1, computing the probability $p_{kp}((U_i,U_j),d)$ of sharing d($1<d<r+w$) key pools between any two nodes differently from U_i and U_j, in here,$i<j$.

When shared key pool contain home key pool S_j of U_j:

$$p_{kp}((U_i,U_j),S_j,d)=\frac{C_{r+w-j}^{d-1}C_{r+w-j-(t-1)}^{w-(d-1)}C_{r+w-i-(w+1)}^{w-d}}{C_{r+w-i}^{w}C_{r+w-j}^{w}}=\frac{C_{r+w-j}^{d-1}C_{r+w-j-t+1}^{w-d+1}C_{r-i-1}^{w-d}}{C_{r+w-i}^{w}C_{r+w-j}^{w}} \tag{3}$$

When it don't contain S_j:

$$p_{kp}((U_i,U_j),\overline{S},d)=\frac{C_{r+w-j}^{d}C_{r+w-j-d}^{w-d}C_{r+w-i-(w+1)}^{w-d}}{C_{r+w-i}^{w}C_{r+w-j}^{w}}=\frac{C_{r+w-j}^{d-1}C_{r+w-j-t}^{w-d}C_{r-i-1}^{w-d}}{C_{r+w-i}^{w}C_{r+w-j}^{w}} \tag{4}$$

When $i=j$:

$$p_{kp}((U_i,U_j),i=j,d)=1 \tag{5}$$

Obtained from formulas (3), (4), (5):

$$p_{kp}((U_i,U_j),d)=\begin{cases}\dfrac{(C_{r+w-j}^{d-1}C_{r+w-j-d}^{w-d+1}+C_{r+w-j}^{d}C_{r+w-j-d}^{w-d})C_{r-i-1}^{w-d}}{C_{r+w-i}^{w}C_{r+w-j}^{w}}, i<j\\ 1, i=j\end{cases} \tag{6}$$

From formula (6), the probability to share at least a key pool between two deployment groups is able to be known:

$$p_{kp}(U_i, U_j) = \sum_{d=1}^{w+1} p_{kp}((U_i, U_j), d) \tag{7}$$

From the initialization, when *i=j*, there is a shared pool at least between two groups:

$$p_{kp}(U_i, U_j) = 1, i = j \tag{8}$$

When $i > r\text{-}w$, U_i share at least a key pool with U_j :

$$p_{kp}(U_i, U_j) = 1, i \geq r - w \tag{9}$$

When $i<r\text{-}w$, formula (6) can be simplified into:

$$p_{kp}(U_i, U_j) = 1 - \frac{C_{r+w-j}^{w} C_{r+w-i-(w+1)}^{w}}{C_{r+w-i}^{w} C_{r+w-j}^{w}} = 1 - \frac{C_{r-i-1}^{w}}{C_{r+w-i}^{w}}, i < r - w \tag{10}$$

From formula (8), (9), (10), (8) can be simplified into (11):

$$p_{kp}(U_i, U_j) = \begin{cases} 1 - \dfrac{C_{r-t-1}^{w}}{C_{r+w-t}^{w}}, i < j < r - w \\ 1, (i = j) \parallel (j > i > r - w \end{cases} \tag{11}$$

According to condition 2, if sensor node shares a key pool with its neighbor node, the probability of sharing key between two nodes is:

$$p_k = 1 - \frac{C_S^m C_S^{S-m}}{(C_S^m)^2} = 1 - \frac{C_S^{S-m}}{C_S^m} = 1 - \frac{((S-m)!)^2}{(S-2m)!S!} \tag{12}$$

There are more keys in key pool, it means S is larger. Therefore, we can calculate p_k approximate from formula (12):

$$p_k \approx 1 - \frac{(1 - \frac{m}{S})^{2(S-m+\frac{1}{2})}}{(1 - \frac{2m}{S})^{2(S-2m+\frac{1}{2})}} \tag{13}$$

Formula (13) shows how the size S of key pool and the one m of key ring influent probability p_k.

We analyzed condition 1 and condition 2 individually and consider synthetically combined them to analyze the probability of shared keys between two neighbor nodes. At last, we concluded, the probability is $1\text{-}(1\text{-}p_k)^d$ below the premise that the two neighbor nodes share d key pools with each other. So, the probability that two nodes share d key pools and key is:

$$\sum_{d=1}^{w+1} p_{kp}((U_i, U_j), d)(1 - (1 - p_k)^d) \tag{14}$$

Stipulating deployment group U_t has deployed in networks at time t, the probability that the nodes of group U_t has connected safely with other nodes deployed in networks is:

$$p_c(t) = \frac{\sum_{i=1}^{t}\sum_{d=1}^{w+1} p_{kp}((i,t),d)(1-(1-p_k)^d)}{t} \quad (15)$$

We can conclude from formula (13), when the size of nodes' key ring is too small relative to the scale of key pools, it will be more probable in security connectivity between nodes. So might make p_k=1 as well:

$$p_c(t) = \frac{\sum_{i=1}^{t} p_{kp}(i,t)}{t} \quad (16)$$

Formula (16) shows, either increasing *w* or decreasing r are able to improve the probability $p_c(t)$ of safe connectivity, the larger the value of w/r, which is greater. During dividing the polynomial keys into groups, it will improve the probability of nodes. When groups is deploying and U_1 has complete, the nodes' safe connectivity probability will peak. It will go through a rapid decreasing, after that, it will go up gradually. Thus after nodes are going to be deployed, the probability will peak on the stage. Therefore, to guaranty the safe connectivity probability between neighbor nodes, before network deployment, we should design proper division for nodes and choose proper r and w to design polynomial grouping.

Be similar with E-G, Erdos's theory about random graph theory is applied to this paper. We take advantage of the random graph theory to discuss the relation between safe connectivity probability of nodes and network safe connectivity. In the theory, the appearance of sides is probability event, and two nodes connect with each other with independent probability p. If the number of sensor nodes is n and the scale of network is bigger, in wireless sensor network, network will be connected graph with probability p_r at least, like that:

$$d = \frac{n-1}{n}(\ln n - \ln(-\ln p_r)) \quad (17)$$

Among the formula, *d* represents the degree of nodes, which is the number of nodes, which connect safely. If the number of neighbor nodes that sensor nodes expect or within the range of nodes' communication is n' $(n'<<n)$, like that:

$$p_e = \frac{d}{n'-1} \quad (18)$$

We can learn from the formula (17) that we should guarantee the degree of nodes for ensure the connected probability in network, that means we should guarantee there are a large number of neighbor nodes connected safely. Also, by the formula (18), we could ensure the size of d through the two ways: (1) improving the probability to connect two nodes safely; (2) increasing the number of expected neighbor nodes, that is increasing density when it deploys in network.

4.2. The Performance of Resistance to the Node Capture

According to the characteristic about designing grouping and deployment of sensor nodes, the process to deploy all the nodes can be divided into *r* intervals, T_1, T_2, …, T_r. As shown in figure 1, every interval T_i can be divided into T_{int} that the process to discover shared keys, T_{del} that the process to delete redundant polynomials and T_f that idle state to deploy. It should cost some time to capture a node when nodes are deployed in networks.

Both discovering keys and deleting redundant keys continue shortly. The sum $T_{int}+T_{del}$ is too small to be captured by attackers. So in our opinion, attacker capture nodes successfully mainly during the idle state T_f of deployment.

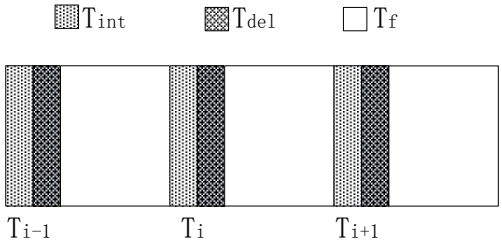

Figure 1. Time Distribution of Nodes' Deployment

Once wireless sensor network is deployed into monitoring area, it will be faced with the variety of attack from aggressor. The paper proposes the scheme to manage keys in networks. Owing to nodes transmit information by encrypting with communication keys, so the networks encounter the main attack that is the node capture. Attacker obtains keys' information stored in nodes by the means of capturing sensor nodes in networks. In this paper, we mainly analyse the probability of compromising polynomial keys after a certain amount of sensor nodes are captured. At time t, when a set number of sensor nodes were captured, we are calculating the key's polynomial $f_{ij}(ID_u, y)$.We might as well suppose u come from the group U_i,in $f_{ij}(ID_u, y)$,j represents polynomial key is from key pool S_j.

Firstly, when node u was captured, the probability that key ring of u contains S_j is:

$$a(u,t)=\begin{cases}0 & , j\le t\\ \dfrac{w(w+r-t)}{w+r-i} & , j>t\end{cases} \tag{19}$$

In that way, when any node was captured, at time t, the expected probability about compromising any key pool is:

$$Ep_{kp}(t)=\frac{\sum_{i=1}^{t}\sum_{j=1}^{r+w}p_{kp}(u,S_j)}{t(r+w)} \tag{20}$$

And after any node was captured, the probability that the key ring contains polynomial $f_{ji}(ID_u, y)$ is:

$$Ep_f(t)=Ep_{kp}(t)\frac{m}{S} \tag{21}$$

That whether the bivariate polynomials preserved in nodes is safe has bearing on its order l. If polynomial's order is l, the polynomial keys' safety will be l-1-order.That means when the number of sensor nodes captured was l, who contains the polynomial, it would be broken to be captured. At time t, while x nodes was captured in network, the probability that there are y nodes containing polynomial f_i would be:

$$p_{f_i}(y,t)=C_x^y(Ep_f(t))^y(1-Ep_f(t))^{x-y} \tag{22}$$

and at time t, the probability that polynomial key f_i was captured would be:

$$p_{f_i}(t)=1-\sum_{y=0}^{l-1}p_{f_i}(y,t) \tag{23}$$

Calculate expectation on time t, in the formula (22), then work out the formula (24), which shows the proportion of polynomial key-compromise with different number of captured nodes.

$$p_{f_i}(x) = \sum_{i=1}^{t} \frac{p_{f_i}(t)}{t} \tag{24}$$

4.3. The Storage Space of Polynomial Keys

In the scheme proposed in this paper, because deployment groups take part in networks, so the one has been deployed into networks needs to go through the state to delete redundant keys. And the number of polynomial keys of sensor nodes' key ring will decrease constantly. At time t, when deployment group U_t's nodes are deployed into network just now, in U_t, there is largest number of keys in key ring during the cutty time discovering shared key polynomial. The home key pool S_t is deleted from nodes, after redundant polynomial keys are deleted. As the subsequent deployment groups join the network, the key pools' number reduces constantly. Then at time t, when all the groups were deployed in networks, the polynomial keys' number reduces to 0.Due to the short time at the discovering shared polynomial keys stage and deleting redundant polynomial keys stage, we mainly concern about the polynomial keys' storage space while networks are running normally.

Because the storage space changes continuously, we should analyze how much keys' storage do sensor nodes cost at different time. We suppose there is a sensor node u in deployment group $U_i(1 \le i \le t)$.At time t, we describe the number of key pools in key ring with $a(u,t)$:

$$a(u,t) = \begin{cases} 0, i < t < r \\ \dfrac{w(w+r-t)}{w+r-i}, t = r \end{cases} \tag{25}$$

We believe that there is same number of sensor nodes in every deployment group. And the expectation for *a(u,t) is:*

$$E_{a(t)} = \frac{\sum_{i=1}^{t} a(u,t)}{t} \tag{26}$$

Ea(t) represents, at time t, the average number of key pools of sensor nodes. During the key pre-distribution stage, every sensor node chooses m polynomial keys from management key pools. So, the key ring averagely contains some polynomial keys whose quantity is represented with *k*(t):

$$k(t) = mE_{a(t)} \tag{27}$$

By the formula (1) introduce the binary polynomial, a l-order polynomial key needs l+1 times storage space of common key. Therefore, at time t, the average storage space of sensor nodes in network is described as:

$$mc(t) = m(l+1)E_{a(t)} \tag{28}$$

In Figure 2, we described how keys' average storage overhead change at different time according to the formula (28). By the Figure 2, it shows that the spending reduces continually as time t pass, because of deleting redundant keys. After deploying sensor nodes first time, that is to say group U_1 was deployed into networks, mc peaked during the whole application cycle, and the average memory spending decreased constantly with

time t passed. At time r, sensor nodes removed all keys, so the keys' memory overhead was 0. Because sensor nodes allocate polynomial keys in key pre-distribution, in this article, it costs more storage overhead of nodes compared with that in literature [15]. However, it greatly improves the network performance. With the process of deployment, the storage overhead continuously decrease to 0, and the storage space occupied by polynomial keys is released to save data of sensor nodes.

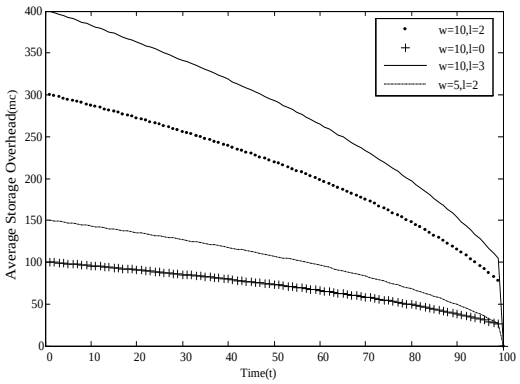

Figure 2. Average Storage Overhead for Sensor Nodes at Different Time

4.4. Node's Computational Overhead

Through this paper, we adopt polynomial keys. The spending for polynomial key correspond to its polynomial computational complexity. The computational process includes two steps:

Step 1: During the polynomial key pre-distribution process, it is necessary to bring sensor node's ID into the polynomial which is allocated randomly. The polynomial is:

$$f(x,y)=\sum_{i,j=0}^{l} a_{ij}x^i y^j = a_{00}+a_{01}x^0y^1+...+a_{ij}x^iy^j+...+a_{ll}x^ly^l$$

The node saves a unary quadratic polynomial of degree l

$$f(y)=a_0+a_1y^1+...+a_jy^j+...+a_ly^l$$

This procedure should take $l(l+1)/2$ times multiplication and l times addition operation. Therefore, the main computation overhead cost to take $O(l)$ times multiplication.

Step 2: During the procedure to discover shared polynomial keys, it is also necessary to take node's ID in $f(y)=a_0+a_1y^1+...+a_jy^j+...+a_ly^l$, and it mainly cost $O(l)$ times to multiply.

While nodes are distributing polynomial keys, based station deposit polynomial keys in sensor nodes through step (1). Hence, there is no overhead in step (1).Though the step (2) spend more computation overhead, because this step decides the main computational complexity of polynomial keys, which is far less than that in asymmetric public-key cryptography algorithms. Besides, the neighbor nodes will get communication key if only they calculate polynomial one time.

4.5. Average End-to-End Delay

In this simulation, 50 nodes move freely in the rectangular region of 500m × 500m , the velocity distribute within the range of 0-20m/s. 512B of CBR data packets were sent

to other nodes by 20 and 40 nodes which is randomly generated. Each node residence time is set to 0s (not stay), 10s, 20s, 40s and 100s.

Figure 3 and 4 respectively compared the end-to-end delay between SMRCL, MODVWLS [16] and AODV [17] protocol in different residence time and different source nodes number. It can be seen from Figure that SMRCL is better than MODVWLS and AODV, which is because SMRCL protocol reduces the Hello message and the network load, also uses the node information of the MAC layer, judge if a node is busy, so that data flow smoothly shunt to the load node and a large number of data streams to timely reach the destination node. In figure 3, the maximum of SMRCL and MODVWLS average delay gap is about 53%, but in figure 4, it becomes 87%, this is because of the increase of the transmission source, SMRCL protocol has more obvious ability balancing node's load in the network.

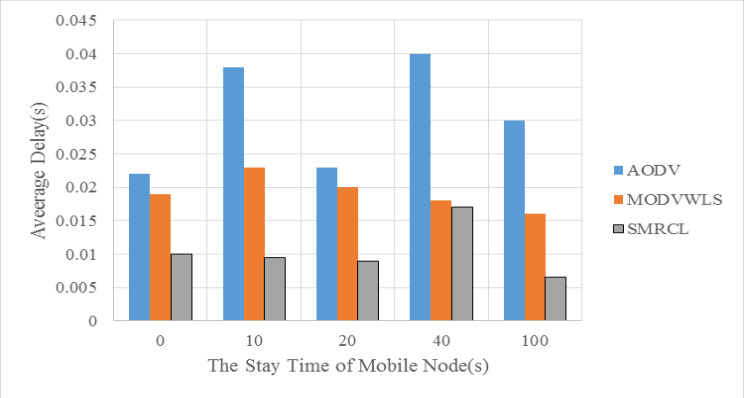

Figure 3. Average End-to-End Delay Comparison (20 Transmission Source)

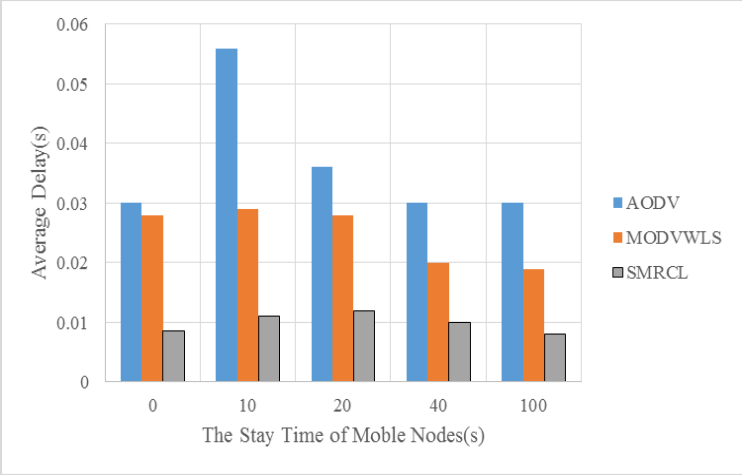

Figure 4. Average End-to-End Delay Comparison (40 Transmission Source)

4.6. Normalized Routing Load

Figure 5 and 6 respectively compared the normalized routing load between SMRCL, MODVWLS [16] and AODV protocol in different residence time and different source nodes number. It can be seen from the figure that SMRCL is better than MODVWLS and AODV under the conditions of 20 and 40 sources. SMRCL routing load is largest decreased by nearly 17% than MODVWLS when the number of transmission sources is 20. SMRCL routing load nearly 41% less than the maximum time of MODVWLS when

the number of transmission sources is 40. This is because MODVWLS does not make improvements for the routing load while SMRCL reduce the Hello messages in the routing process, to avoid a large number of invalid information. Secondly, due to the reduction of the Hello message, the link detection between nodes reduces, avoiding frequent issued RREQ. And finally SMRCL limits the RREP returns of the non-destination node, reducing a large number of invalid routing information.

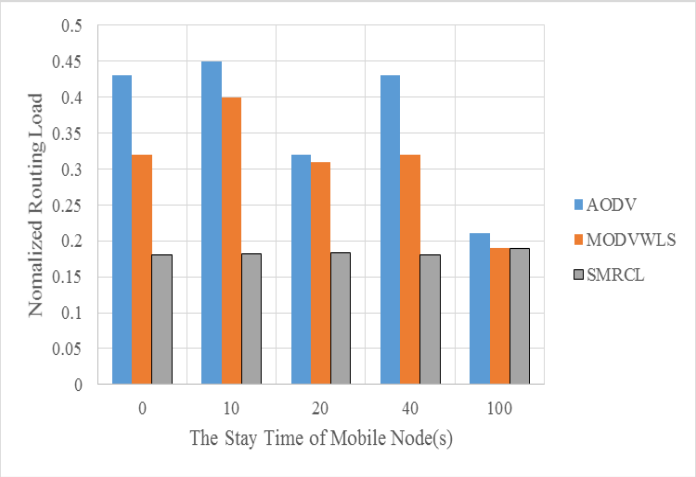

Figure 5. Normalized Routing Load Comparison (20 Transmission Source)

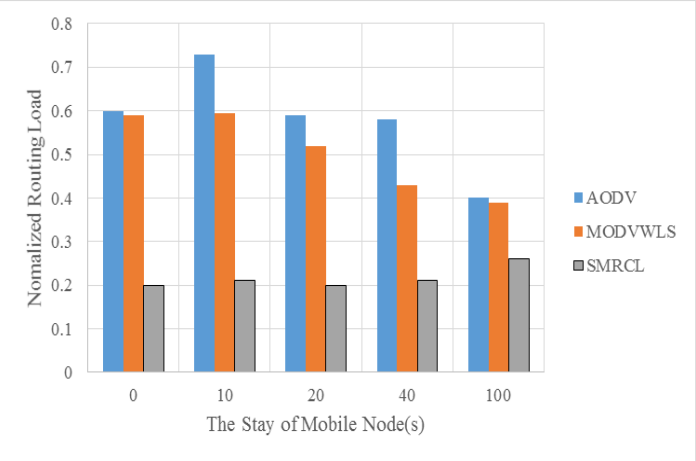

Figure 6. Normalized Routing Load Comparison (40 Transmission Source)

We get that the SMRCL can reduce energy consumption, increase network throughput, reduce the average delay though simulation analysis above.

5. Conclusions

In this paper, a key management and multi-hop routing scheme based on cross layer was proposed for wireless sensor networks. The core of SMRCL has two parts as following. Firstly, this paper proposes a key management scheme using polynomial and time-based deployment, which is on the basis of group-divisible design for sensor nodes and key pools and introduce bivariate symmetric polynomial to replace common keys. Group-divisible design reduces the spending on nodes, and using polynomial to replace

common keys enhances the wireless sensor network resilience against capture. At the initialization phase, respectively divide sensor nodes and key pools into groups in accordance with network size and safety requirements. The dynamic deployment improves the expansion of network performance. At discovering polynomial keys phase, because the scheme encrypt KeyID with pre-deployed one-way key to prevent KeyID leak, hence attackers will not attack on sensor nodes. While deleting the redundant polynomial keys lower the probability to compromise keys, it reduces the storage overhead also. Secondly, a multi-hop routing algorithm proposed based on cross-layer, which can be used in large scale. In the algorithm, we can use the information of the MAC layer, consider the cross-layer design, rich nodes busy judgment and improve the network load ability. The MAC layer provides two parameters: the ratio of the packet sending rate and packet reception rate and queue length. As the analysis, the separate use of the data packet reception rate and packet sending rate ratio can't be good to determine whether the node is in busy state. So at the same time it uses the queue length to represent the actual size of the current node's buffer. The queue length represents the packet length stored in node buffer at some time point. Simulation results demonstrate that it can save energy and balance the network load and obvious improve the network lifetime.

Acknowledgements

The author would like to thank the Chongqing Natural Science Foundation under Grant No. cstc2012jjA40038 and the Chongqing Basic and Frontier Research Project under Grant NO.cstc2013jcyjA40023, cstc2014kjrc-qnrc40002. The work presented in this paper was supported in part by the Ministry of Industry and Information Technology of the Peoples Republic of China for the special funds of Development of the Internet of things (2012-583).

References

[1] Ren FY, Huang HN and Lin C., "Wireless sensor networks [J]", Journal of Software, vol. 14, no. 7, **(2003)**, pp. 1282-1291.
[2] Li JZ, Li JB and Shi SF, "Concepts, issues and advance of sensor networks and data management of sensor networks [J]", Journal of Software, vol. 14, no. 10, **(2003)**, pp. 171-172.
[3] S. Fengjun, "Communication Protocol for wireless sensor networks [M]", Publishing House of Electronics Industry, **(2011)**, pp. 17-25.
[4] Z. Yaqin and T. Dingzhong, "The applications and research status of wireless sensor network [J]", Sensor World, vol. 5, **(2009)**, pp. 35-40.
[5] F. Tao, W. Yilin and M. Jianfeng, "A New Ad hoc Group Key Agreement Scheme Based on ECC [J]", Chinese Journal of Electronics, vol. 37, no. 5, **(2009)**, pp. 918-924.
[6] S. Wen, "The Technology and Application of wireless sensor network [M]", Beijing: Publishing House of Electronics Industry, **(2007)**, pp. 1-29.
[7] W. Jiang-Hong, L. Wen-Fen and H. Xue-Xian, "Provable Secure Attribute Based Authenticated Key Exchange Protocols in the Standard Model [J]", Journal of Software, vol. 25, no. 10, **(2014)**, pp. 2397-2408.
[8] Ma HD and Tao D., "Multimedia sensor network and its research progresses [J]", Journal of Software, vol. 17, no. 9, **(2006)**, pp. 2013-2028.
[9] Y. wangke, M. wenping, Chehefeng and G. sheng, "Efficient key management scheme for wireless sensor networks [J]", Joural of Southeast University (Natual Science Edition), vol. 41, no. 1, **(2011)**, pp. 20-24.
[10] T. Hong, J. Xie and Y. Lu, "The Principle and Application of wireless sensor networks [M]", Beijing: Post &Telecom Press, **(2010)**, pp. 20-22.
[11] W. Xiaogang, Weiren and Z. Wei, "A key management scheme based on quadratic form for wireless sensor network [J]", Chinese Journal of Electronics, vol. 41, no. 2, **(2013)**, pp. 214-219.
[12] L. M. L. Oliveira and J. P. C. Rodrigues, "Wireless sensor networks: a Survey on Environmental Monitoring [J]", Journal of Communications, vol. 6, no. 2, **(2011)**, pp. 143-151.
[13] Guo JF, Zhang XM, Xie F and Chen GL, "A leisure degree adaptive routing protocol for mobile ad hoc network [J]", Journal of Software, vol. 16, no. 5, **(2005)**, pp. 960-969.
[14] S. Cheng, L. Yi-fei and X. Qin, "An Integrated Metrics Based Routing Protocol for Wireless Mesh Networks [J]", Chinese journal of Computers, vol. 33, no. 12, **(2010)**, pp. 2300-2311.

[15] Yuan T., Ma JQ, Zhong YP and Zhang SY, "Key management scheme using time-based deployment for wireless sensor networks [J]", Journal of Software, vol. 21, no. 3, **(2010)**, pp. 516-527.
[16] F. yunqing, W. songjian and W. zhongfu, "A Routing Protocol of Wireless Mesh Network Based on Weighted Link State [J]", Journal of Computer Research and Development, vol. 46, no. 1, **(2009)**, pp. 137-143.
[17] Perkins CE and Royer EM, "Ad-hoc on-demand distance vector Routing[C]", Mobile Computing Systems and Applications, **(1999)**, pp. 90-100.

Authors

Fengjun Shang (1972-), male, finished his Ph.D. degrees in Instrument Science and Technology at the College of Opto-electronic Engineering, Chongqing University, China, in 2005. Since then he works at the Institute of Computer Network Engineer in Chongqing University of Posts and Telecommunications, China. He was a visiting scholar in University of Wollongong, Australia, from November 2007 to November 2008. His research interests include sensor network, IOT, network optimization and Cloud Computing.

