Key Management Scheme Based on Nodes Capture Probability for Wireless Sensor Networks

Ying Zhang, Peisong Li

College of Information Engineering, Shanghai Maritime University, Shanghai 201306
E-mail: yingzhang@shmtu.edu.cn

Abstract: The security of wireless sensor network is becoming more and more important, in which key management is an important part to realize the security of the networks. In the hierarchical network, cluster heads play an important role in the network. Once being captured, they will expose more keys and information. Therefore, how to choose the cluster head should be considered seriously. However, the existing clustering algorithms only consider energy factor when selecting cluster heads. This paper proposes a model regarding probability of nodes being captured which not only considers the energy factors, but also considers the nodes capture probability when choosing cluster heads, so that nodes which hold smaller capture probability tend to be the cluster heads. On the basis of this model, an efficient key agreement scheme is proposed, which can use EBS (Exclusion Basis Systems) structure to update the group keys between clusters effectively. Simulation experiments show the proposed scheme has good connectivity and node capture resistance, and it can also save more storage cost.

Key Words: Wireless sensor networks, Key management, Nodes capture probability, EBS

1 Introduction

Wireless sensor networks (WSNs) are intelligent network systems which are capable of data acquisition, fusion and transmission [1]. WSNs have important scientific research value and huge market value in many fields, such as space exploration, family health monitoring, environmental monitoring, urban traffic management, freight management, military reconnaissance, warehousing management and so on. However there are still many security problems needed to be solved in WSNs. Key management can ensure the security communication of the networks, which involves all kinds of management problems of keys, such as key generation, allocation, transmission, revocation and so on. Efficient key management is also the basis of ensuring the effective operation of other security mechanisms, such as security data fusion, security positioning, secure routing, etc. However, due to the limited energy and lower computation capability of nodes in WSNs, the complex key management schemes are not suitable to be used in WSNs.

According to the different topology structures, WSNs can be divided into the planar network and the layer cluster network. The layer cluster WSN has the advantages of high energy utilization, and many scholars have proposed the routing protocols based on clustering [2]. Most of these protocols only consider the energy factor and do not pay attention to the security problems. Key management is an important method to ensure the security of clustering. WSN nodes are often deployed in the adverse area for monitoring, and there is danger of being captured by the enemy. In the clustered network, once cluster heads were captured, more keys and information will be exposed, and these will threaten the security of the whole networks. A network model of nodes capture probability is proposed in this paper. The nodes capture probability is considered while clustering.

*This work is supported by the National Natural Science Foundation of China under Grant 61673259, and International Exchanges and Cooperation Projects of Shanghai Science and Technology Committee under Grant 15220721800.

2 Related Work

Key management is an important part in the technology of data encryption. Nowadays, many key management methods have been proposed. Eschenauer and Gligor first proposed the random key pre distribution method (E-G). The idea is that the server generates a large key pool, all nodes hold some key in the key pool, so long as there is the same key between the adjacent nodes, a secure channel can be built. Camtepe et al. proposed a key management method based on region combination design [3], and its key connectivity rate could remain higher.

Many scholars proposed layer cluster key management of WSN based on the hierarchical network structure [4]. Jolly et al. proposed an energy-efficient key management protocol [5], which is built on the hierarchical network structure, and it effectively reduces the energy consumption caused by key management. Zhu et al. proposed the LEAP key management method [6], which supports the passive participation and the intra network processing, and provides authentication of intra network nodes. Afterwards, on the basis of the LEAP method, the scholars also proposed some other methods which have better performances, such as LEAP+, OTMK, EDDK and so on. In heterogeneous hierarchical network, Du et al proposed a key pre-distribution method [7]. Compared with the existing random key pre-distribution methods, it enhances the ability of anti-capture, but the key storage cost is large, and the cluster cannot be changed dynamically. On the base of EBS [8], Younis et al. proposed a dynamic key management: SHELL based on the locations [9]. SHELL has the ability to resist on collusion attack better than those of random key distribution methods, and it has strong expansibility. However, the more the key gateway generation node is captured by the adversary, the greater the probability of exposing secret information will be. Most of the present key management methods are not extensible and cannot adapt to the dynamic changes of the network.

978-1-5386-1243-9/18/$31.00 ©2018 IEEE

3 Network Model

The proposed key management method uses layer cluster as shown in Fig 1, and the following assumptions are made in this network model:

1. The node has a unique ID and is randomly deployed in the monitoring area.

2. The position of all nodes is fixed and the energy is limited.

3. All nodes have the same capability and equal status. Once captured, the stored key will be compromised.

4. The transmit power of the node can be adjusted according to the distance.

5. All the nodes know their location, and can perform data acquisition task periodically.

6. The base station is safe, reliable and has sufficient energy. It can communicate with any node in the monitoring area, and after a certain time, the base station can re-cluster the network.

7. Intrusion detection technology is used in the network, captured nodes can be found by base station.

This paper also proposes a network model of node capture probability as formula (1):

$$G=\begin{cases}0 & 0\le d<d_0\\ G_{\max}\times(\dfrac{d-d_0}{d_H-d_0})^2 & d_0\le d<d_H\\ G_{\max} & d_H\le d\end{cases} \tag{1}$$

where d_H is the threshold value of the monitoring area, d_0 is the safety value of the monitoring area, if the distance from the base station is less than d_0, the probability of a node being captured is almost 0, and $G_{\max}$ is the maximum capture probability. d_H , d_0 , $G_{\max}$ are chosen according to the specific monitoring environment, d is the distance from the node to the base station. If the location of the base station is the safest, when the distance from the base station is less than d_0 , as a result of far away from the enemy area, the probability of the enemy appearance is almost 0, node is difficulty to be captured, so the node has the minimum capture probability. When exceeding d_0 , the chance of enemy appearance will be increasing, and the nodes capture probability will also be increased. When the distance from the base station exceeds the threshold d_H , the node needs to monitor the adversary's main active area, and the node has the maximum capture probability.

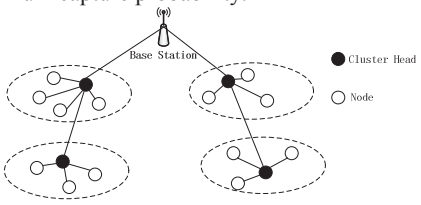

Fig 1. Network model of the layered cluster WSNs

4 Clustered Multi-hop Routing Algorithm based on Nodes Capture Probability

In this paper, the energy factors and the node capture probability are considered, the particle swarm optimization (PSO) is used to cluster the network, and the multi-hop mode is used to transmit data between the clusters.

4.1 Particle Swarm Optimization

PSO algorithm [10] is an iterative optimization algorithm, which initializes the particle swarm into a set of random solutions, and all particles follow the current optimal particle to search the optimal solution in the solution space. During the iteration, the particles are updated according to 2 extremums, namely their own individual extremum and global extremum. Among them, the individual extremum is the optimal solution found by the particle, and the global extremum is the optimal solution found by the entire particle swarm. The update formula for the particle position and velocity are as follows:

$$\begin{aligned}v_{id}(t)&=w\times v_{id}(t-1)+c_1\times\varphi_1(p_{id}-x_{id}(t-1))+c_2\times\varphi_2(p_{gd}-x_{id}(t-1))\\ x_{id}(t)&=x_{id}(t-1)+v_{id}(t)\end{aligned} \tag{2}$$

where v is particle velocity, x is particle location, c_1, c_2 is learning factor, φ_1, φ_2 is the random number between 0 and 1, p_{id} is the optimal location of particles, p_{gd} is the global optimal position, $pbest_i$ is the fitness value corresponding to the p_{id} $gbest$ is the fitness value corresponding to the p_{gd}, and *w ls the* inertial weight.

4.2 Cluster head selection algorithm based on Particle Swarm

The implementation of the protocol adopts a polling mechanism, and each round includes two stages: the establishment of clusters and the steady state of transmission. After the nodes are deployed, their position and energy information are sent to the base station. Because the base station knows the initial energy of the nodes, the energy consumption of each node can be estimated by the clustering information of each round, and the energy information of node in each round can be obtained. Because the node position is fixed, there is no need to send location and energy information to the base stations in the subsequent rounds; or send the location and energy information to the base station within a longer period of time. Considering the nodes capture probability while clustering, the fitness function is set as follows:

$$f=\beta_1\times f_1+\beta_2\times f_2+\beta_3\times f_3 \tag{3}$$

where

$$f_1=\max_{k=1,2\cdots K}\left\{\sum_{\forall n_i\in C_{p,k}} d(n_i,CH_{p,k})/|C_{p,k}|\right\} \tag{4}$$

$$f_2=\sum_{i=1}^{N}E(n_i)\Big/\sum_{k=1}^{K}E(CH_{p,k}) \tag{5}$$

$$f_3=\sum_{k=1}^{K}A\times G_k\Big/K \tag{6}$$

f_1 is the clustering compactness evaluation factor, which equals to the maximum average Euclidean distance to the corresponding cluster head node; f_2 is cluster head energy evaluation factor, which equals to the sum of all nodes $n_i(\mathrm{i}=1,2,\cdots \mathrm{N})$ energy in the network divided by the sum of the current cluster head energy; f_3 is cluster head capture evaluation factor. $d(n_i,CH_{p,k})$ is the distance from the node n_i to the cluster head, $|C_{p,k}|$ is the number of nodes of cluster C_k in particle P; G_k is the probability of each cluster head being captured. β_1,β_2,β_3 is weight coefficient of each evaluation factor, and $\beta_1+\beta_2+\beta_3=1$. The specific steps are as follows:

1) Initialize the position and velocity of each particle $P(P=1,2,\cdots S)$, each contains K candidate cluster heads.

2) Assign the node n_i to the nearest cluster head, according to (4) - (6), calculate the fitness of each particle.

3) The individual optimal solution of each particle $pbest_i$ and the general optimal solution of the population $gbest$ are determined.

4) Update the particle velocity and location.

5) Map the particle position to the cluster head node's location.

6) Repeat step 1 - 5 until the maximum number of iterations is reached.

After the cluster is set up, the system enters to the steady state phase. After the cluster head receives the information sent by all nodes in the cluster, the cluster head fuses the received raw data, and then transmits the fused data to the base station by multi-hops.

4.3 The Inter-cluster multi-hop routing

The intra-cluster node adopts single-hop transmission mode to send information to the cluster head. Instead, the multi-hop transmission mode is adopted among the clusters. The design goal of inter-cluster multi-hop routing is to find an optimal path to reduce the energy consumption of inter-cluster data transmission. Only meeting the formula (7), the multi-hop transmission can save energy.

$$d^2_{CH_i-CH_j} + d^2_{CH_j-BS} < d^2_{CH_i-BS} \quad (7)$$

The cluster head CH_i selects relay nodes in all cluster heads according to the weight value, and the weight function is defined as formula (8).

$$W(i,j) = \frac{S(j).E}{E\_everger(r-1)} + \frac{d^2_{CH_i-BS}}{d^2_{CH_i-CH_j} + d^2_{CH_j-BS}} \quad (8)$$

The weight size is used as the selection basis of the next route, the cluster head with the largest weight value is used as the cluster head node of the next hop data transmission, and the data is directly transmitted to the base station if the weight value of the cluster head is the maximum.

5 KMNCP key Management Method

Considering the existing key management methods do not fully consider the dynamic update keys and the capture probability of cluster head, this paper proposes a key management method based on nodes capture probability (KMNCP).

5.1 Key Establishment

Supposing that N nodes are safe in the beginning of the deployment, and at this period of time they will not be captured by the adversary, each node stores an initial key K_0, a one-way hash function H, the unique id of the whole network and the base station id_{BS}. Nodes are randomly deployed in the monitoring area, each node n_i sends location L_i, energy information E_i and id_i to the base station through key K_0:

$$n_i \rightarrow BS : E_{K_0}(id_i \,||\, L_i \,||\, E_i) \quad (9)$$

After the base station receives the information of each node, considering the capture probability of the node, the method mentioned above is used for clustering, broadcast cluster head id_{C_j} and position information L_j after selecting K cluster head at base station:

$$BS \rightarrow * : E_{K_0}(id_1 \| \cdots id_K \,||\, L_1 \,||\, \cdots L_K) \quad (10)$$

After each decryption node decrypts the received information, then the nearest cluster head is added according to the location, and separately calculates the session key and cluster key that communicate with the cluster head and the base station, after the three session keys are established, the initial key K_0 will be deleted. Suppose node u needs to establish a session key with the cluster head:

$$K_{uCH_j} = H(K_0 \,||\, id_{CH_j} \,||\, id_u) \quad (11)$$

$$K_{uBS} = H(K_0 \,||\, id_{BS} \,||\, id_u) \quad (12)$$

$$K_{CH_j} = H(K_0 \,||\, id_{CH_j}) \quad (13)$$

After receiving the information, the cluster head calculates the session key communicating with the base station and the session key communicating with the other cluster heads:

$$K_{CH_jBS} = H(K_0 \,||\, id_{BS} \,||\, id_{CH_j}) \quad (14)$$

$$K_{CH_jCH_i} = H(K_0 \,||\, id_{CH_j} \,||\, id_{CH_i}) \quad (15)$$

Then K_0 will be deleted. The base station obtains the session key communicating with all the nodes and the cluster key of each cluster based on the formula (11) - (14), Then the node information assigned to the cluster head is transferred to the cluster head through the K_{CH_jBS}, the EBS structure is constructed among the clusters, and the shared key K_{CH} between the cluster heads is sent to each cluster head:

$$BS \rightarrow CH_j : E_{K_{CH_jBS}}(K_{ebs} \,||\, K_{CH} \,||\, m) \quad (16)$$

where K_{ebs} is the EBS key management set, m is the common node's related information allocated to the cluster head. After each cluster head gets the information of the intra-cluster node, the session key to communicate with the intra-cluster node is calculated according to formula (11), and the whole network key is established.

5.4.2 Key establishment when polling cluster

Assuming that the cluster establishment time T is less than the nodes capture time $T_{\min}$. In the cluster phase, the base station uses the PSO algorithm to select K cluster heads. Then, the selected cluster head information and key update parameter S are broadcast by using the cluster key of each cluster in the previous round, and the new cluster head information and S are broadcast to the cluster head in the last round by using the shared key K_{CH} between the cluster heads. After decrypting the received information, each non-cluster head node selects the nearest cluster head to join according to the location, calculates the session key and the cluster key communicating with the cluster head, and updates the session key communicating with the base station. After the key establishment is finished, delete S. Suppose node u needs to establish a session key with the cluster head:

$$K'_{uCH_j} = H(S \,||\, id_{C_j} \,||\, id_u) \quad (17)$$

$$K'_{uBS} = H(K_{uBS} \,||\, S) \quad (18)$$

$$K'_{CH_j} = H(S \,||\, id_{CH_j}) \quad (19)$$

The base station transmits the assigned common node's information, K'_{CH} and K'_{ebs} to the cluster head through the K_{uBS}:

$$BS \rightarrow CH_j : E_{K_{CH_jBS}}(K'_{ebs} \,||\, K'_{CH} \,||\, m') \quad (20)$$

After each cluster head obtains the information of the intra-cluster node, the session key communicating with the

intra-cluster node is calculated according to the formula (17), and after the key establishment is finished , S will be deleted.

5.2 Key Updating

During the stable transmission phase, if the key is used for too long, it is easy to be cracked by the adversary, so the key needs to be updated regularly. The key update is initiated by the cluster head, and each cluster head computes $P(x)=(x-K_1)(x-K_2)\cdots(x-K_a)$, K_a is the session key between the cluster head and the intra-cluster nodes. Cluster head multicast $g(x)=P(x)+S'$ within the cluster, the intra-cluster node brings the session key to $g(x)$ and the key update parameter S', and then each node calculates the new session key used to communicate with the cluster head:

$$K'_{uCH_j}=H(K_{uCH_j}\,||\,S') \tag{21}$$

The base station updates parameter S' by broadcasting the key via K_{CH}, and the cluster head updates the session key used to communicate with the other cluster heads after receiving the information:

$$K'_{CH_jCH_i}=H(K_{CH_jCH_i}\,||\,S') \tag{22}$$

5.3 Key Revocation

If a common node is captured, and it is found by base station, cluster head will be informed to delete the key associated with the captured node through the communication key to cluster head. If cluster head is captured, base station will send group key update information between cluster heads through the EBS management key set. For a group of 10 cluster heads, you can build $EBS(N,K,M)=EBS(10,3,2)$, Table 1 is a regular matrix $EBS(10,3,2)$, and each cluster head stores 3 keys with a total of 10 different methods. Suppose cluster head CH_1 is captured, due to the possession of the key K_1,K_2,K_3, these exposed keys need to be revoked and updated. $K_4\cup K_5$ is a set of keys owned or partially owned by all members except the node CH_1, so the encryption keys that can be used to update the keys are K_4 and K_5. Therefore, two broadcast information can be constructed, that is:

$$M_1=E_{K_4}(K'_{CH}\,||\,E_{K_1}(K'_1)\,||\,E_{K_3}(K'_3)\,||\,E_{K_5}(K'_5)) \tag{23}$$

$$M_2=E_{K_4}(K'_{CH}\,||\,E_{K_1}(K'_1)\,||\,E_{K_3}(K'_3)\,||\,E_{K_5}(K'_5)) \tag{24}$$

K'_{CH} is the key shared between the new cluster heads, by broadcasting two messages, the new key K'_1,K'_2,K'_3 will replace the exposed key K_1,K_2,K_3, at the same time, only the legitimate member nodes can decrypt the information and can only obtain a new key of its own exposed key.

Table 1: $EBS(10,3,2)$ Regular Matrix

Node key	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
K_1	1	1	1	1	1	1	0	0	0	0
K_2	1	1	1	0	0	0	1	1	1	0
K_3	1	0	0	1	1	0	1	1	0	1
K_4	0	1	0	1	0	1	1	0	1	1
K_5	0	0	1	0	1	1	0	1	1	1

The base station simultaneously let the node with the energy greater than the average energy of the nodes in the cluster and with the smallest capture probability become the new cluster head of the cluster. Since the cluster head does not store the cluster key during the key establishment process, the base station can multicast the captured information of the cluster head, the information m of the new cluster head CH_j and the key update parameter S' to the cluster member through the cluster key:

$$BS\rightarrow *: E_{K_{CH_j}}(m\,||\,S') \tag{25}$$

After receiving the message, the intra-cluster node deletes the session key communicating with the old cluster head, and establishes the session key communicating with the new cluster head:

$$K'_{uCH_j}=H(S'\,||\,id_{CH_j}\,||\,id_u) \tag{26}$$

The base station transmits the information of the nodes in the cluster, K'_{CH} and K'_{ebs} to the new cluster heads through K_{uBS} , the new cluster head will get session key communicating with the intra-cluster nodes according to formula (17). In addition, when a cluster head is captured, the structure of the cluster in the network will not be changed because a new node in the cluster becomes the cluster head of the cluster.

6 Methods performance evaluation

Through simulation and analysis, the key management method proposed in this paper is evaluated in terms of connectivity, storage, communication overhead and security. In the simulation experiment, 200 nodes are assumed to be deployed randomly in the area of $200\times200\text{m}^2$. The communication energy consumption model proposed in [11] is adopted, the energy consumed when the length of the data is k bits and the distance is d can be described as formula (27).

$$E_{Tx}(k,\mathrm{d})=E_{Tx\_elec}(k)+E_{Tx\_amp}(k,\mathrm{d})=\begin{cases}kE_{elec}+k\varepsilon_{fs}d^2 & d<d_0\\ kE_{elec}+k\varepsilon_{mp}d^4 & d\geq d_0\end{cases} \tag{27}$$

where ε_{fs} and ε_{mp} are the energy consumption coefficients of the power amplifier circuit, $d_0=\sqrt{\varepsilon_{fs}/\varepsilon_{mp}}$.

The energy consumption in receiving k bit data is described as formula (28).

$$E_{Rx}(k)=E_{Rx\_elec}(k)=kE_{elec} \tag{28}$$

where $E_{elec}=50nJ/bit$, $\varepsilon_{fs}=10pJ/bit/m^2$, $\varepsilon_{mp}=0.0013pJ/bit/m^4$, data fusion energy consumption is $E_{DA}=5nJ/bit$.

Because the power of the node is adjustable, it is assumed that the node communicates with the base station at a higher power level, the maximum communication radius of normal communication is $d_{\max}=40m$, which means the nodes within the 40m radius are the adjacent nodes. Suppose that the node *ID*, energy information, location information and key identifier are 16 bits, the key length is 64bit, the HELLO information in LEAP is 16 bits, and the MAC information is 64bit.

6.1 Safety Analysis

The proposed KMNCP scheme is based on the hierarchical network structure. The cluster head is the key node in the network, and it contains a large number of keys. Once captured, a large number of keys in cluster head will be

leaked. The reestablishment of the cluster requires a large amount of energy consumption, so that the node with the small capture probability should be made as the cluster head as much as possible while clustering. Considering the nodes capture probability, a multi-hop routing protocol (PSO-GP) based on particle swarm optimization is compared with the multi-hop routing protocol (PSO-MH) based on particle swarm optimization without considering the nodes capture probability and the single-hop routing protocol (PSO-C) based on particle swarm optimization, and the simulation parameters are shown in Table 2. Among them, the selected area is $200\times200\text{m}^2$, and the maximum communication radius is 40m, and the most suitable number of particles could be chosen as 20. In PSO-MH and PSO-C, nodes with more than average energy can be considered as the candidate cluster heads. In PSO-GP, node capture probability is considered on the basis of PSO-MH, and nodes with energy greater than 1/2 average energy are considered as candidate cluster heads. The time of the death of the first node appearance is defined as the lifetime of the network. As shown in Fig. 2, the first death node of PSO-GP appears in 270 rounds, and the first death node of PSO-C appears in the 229 round, and the PSO-MH is in the 282 round. The average capture probability of the cluster heads in every five rounds is calculated as shown in Fig. 3. Because the network is incomplete after the dead node appears, the average capture probability of cluster head will be compared each other before the first dead node appears. In the network survival time of PSO-GP, its average capture probability of cluster heads is lower than PSO-C and PSO-MH over 80% times. That means the cluster head selected by PSO-GP with considering nodes capture probability is more secure. PSO-GP's network survival time and network energy consumption balance are not as good as PSO-MH, and the first dead node of PSO-MH is only 12 rounds ahead of PSO-GP. The network energy consumption balance and network survival time of PSO-GP are still better than single-hop routing protocol PSO-C. PSO-GP sacrifices a little network energy consumption balance, but greatly improves the network security, this sacrifice is acceptable.

The survivability of the key management method refers to the ability of the node to resist on the attack. In other words, it refers to the probability that the shared secret key leakage between other nodes after the node in the network is captured. In the KMNCP method, each node shares a unique session key with the cluster head. When the node is captured, the communication key between other nodes will not be leaked, so the KMNCP method has good survivability. If the node is captured, the probability of sharing key leakage between other nodes is close to 0.

Table 2: Simulation Parameters

Parameter	Parameter value
The number of particles S	20
Learning factor c_1, c_2	2
Inertial weight w	0.9-0.4 changes linearly
Weight coefficient β_1	0.25
Weight coefficient β_2	0.15
Weight coefficient β_3	0.6
The number of Cluster head K	10
Constant A	150
Maximum capture probability $G_{\max}$	0.1
Monitoring area threshold d_H	225m
Monitoring regional safety values d_0	45m
Number of transfers data per round	4000bit
Node initial energy E_0	0.2J

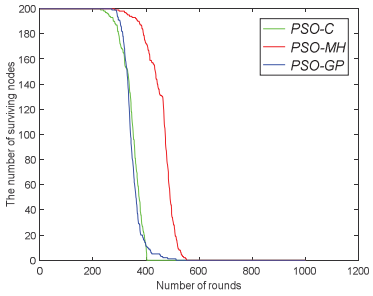

Fig 2. The comparison of network lifetimes

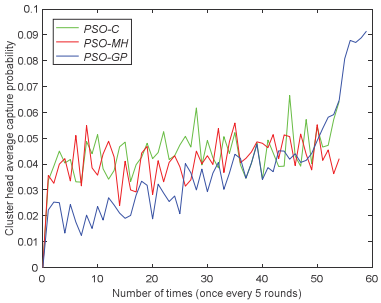

Fig 3. Average capture probability of cluster heads (take the average of the average capture probability of cluster heads every 5 rounds)

6.2 Key Connectivity

Key connectivity refers to the probability that a node can directly establish a communication key between the nodes. In the proposed KMNCP key management scheme, each node can establish a unique session key with the cluster head, and the cluster head also can establish a unique communication key, so the key connectivity in the network is close to 1. LEAP scheme in key connectivity is similar with KMNCP method. In the E-G method, the key connectivity is related to the size of key pool S and the number of keys stored in the node. If you want to get a higher key connectivity rate, each node needs to store enough keys. In the case of S=10000, only when 75 keys are stored in each node, the connectivity rate of 0.5 can be achieved, so the key connectivity rate is low.

6.3 Storage Overhead

The storage space of nodes in WSNs is very limited. In the case of security, the overhead of key storage should be reduced as much as possible. In the proposed KMNCP key management scheme, each common node needs to store two

session keys and one cluster key to communicate with the base station and the cluster head, and the cluster head needs to store the session key to communicate with the intra-cluster nodes, session key for communication with base station and the EBS key set. It is assumed that there are N common nodes and M cluster heads in the network, the EBS management key for each cluster head node is K, and the number of keys stored in KMNCP is $4\times N+K\times M$.

There are four types of keys in the LEAP scheme: group keys (the shared key between all the keys and base station); cluster key (the shared key between the node and all its neighbors); the shared pair key (the shared key between the node and its direct adjacent nodes); private key (the shared key between each node and base station). In LEAP scheme, if a node has d neighbors, regardless of the authentication keys between nodes, the number of four types of keys stored in each node is $2\times d+2$.

The value of K is set as 10 in simulation, $EBS(N,k,m)=(10,3,2)$, that means each cluster head stores 3 EBS key sets . From Fig. 4, we can see that the key storage overhead of the proposed KMNCP scheme is lower than the representative scheme LEAP.

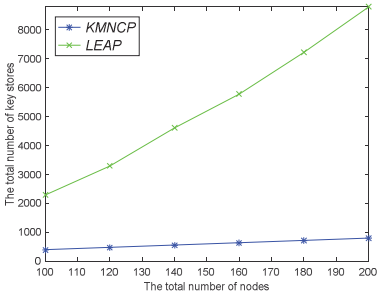

Fig 4. The comparison of key storage overhead

6.4 Expenditure of Energy

The nodes are usually powered by batteries, and the energy is limited, so the energy consumption of key management scheme should be as small as possible when the key is set up. The proposed scheme only needs to receive the key parameter information sent by the base station to get the required communication key by the one-way hash function, so the communication energy consumption is relatively low. Because the computational energy consumption is smaller than the communication energy consumption, the energy consumption of the sensor transmitting 1 bit is about 1000 times of the energy of performing the computational command. So we only consider the communication energy consumption when establishing the security key. Regardless the energy consumption of base station, we can make the comparison of the nodes communication consumption while establishing the keys between KMNCP and LEAP schemes. In the LEAP method, the energy consumption of the four kinds of keys is considered. As seen in Fig. 5, the communication energy consumption of KMNCP is lower while establishing the keys.

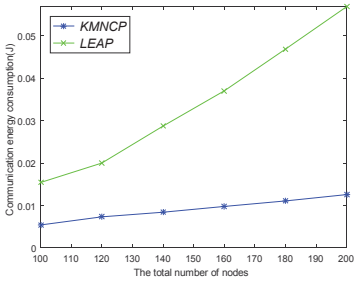

Fig 5. The comparison of communication energy consumption of key establishment

7 Conclusion

A node capture probability network model is proposed. Besides of considering the energy factor, the nodes capture probability is also considered while clustering. The node with more secure has more probability to become a cluster head. The proposed key management scheme effectively reduces the storage cost and energy consumption when the key is established. By using EBS key system, the group key update between cluster heads is realized effectively.

REFERENCES

[1] Yick J, Mukherjee B, Ghosal D. Wireless sensor network survey, Computer networks, 52(12): 2292-2330, 2008.

[2] Waware S, Sarwade D N, Gangurde P. A review of power efficient hierarchical routing protocols in wireless sensor networks, International Journal of Engineering Research and Applications, 2(2): 1096-1102, 2012.

[3] Camtepe S A, Yener B. Combinatorial design of key distribution mechanisms for wireless sensor networks, Computer Security–ESORICS 2004. Springer Berlin Heidelberg, 293-308, 2004.

[4] Xiao Y, Rayi V K, Sun B, et al. A survey of key management schemes in wireless sensor networks, Computer communications, 30(11): 2314-2341, 2007.

[5] Jolly G, Kuscu M C, Kokate P, et al. A low-energy key management protocol for wireless sensor network. In: Proc. of the Eighth IEEE Intl. Symposium on computers and communication (ISCC'03). Kemer-Antalya, Turkey: 3-3 July, 335-340, 2003.

[6] Zhu Sencun, S.Sanjeev, J.Sushi. LEAP: efficient security mechanisms for Large-scale distributed sensor networks, proceedings of the10th ACM Conference on Computer and Communication Security, Washington DC, 62-72, 2003.

[7] Du X, Xiao Y, Guizani M, et al. An effective key management scheme for heterogeneous sensor networks, Ad Hoc Networks, 5(1): 24-34, 2007.

[8] Eltoweissy M, Heydari M H, Morales L, et al. Combinatorial optimization of group key management, Journal of Network and Systems Management, 12(1): 33-50, 2004.

[9] Younis M, Ghumman K, Eltoweissy M. Location-aware combinatorial key management scheme for clustered sensor networks, IEEE Transactions on Parallel and Distributed Systems, 17(8): 865-882, 2006

[10] Clerc, M., & Kennedy, J., The particle swarm: Explosion, stability, and convergence in a multi-dimensional complex space, IEEE Transactions on Evolutionary Computation, *6*, 58–73, 2002.

[11] Xu N. A survey of sensor network applications, IEEE Communications Magazine, 40(8): 102-114, 2002.