RANDOM KEY PRE-DISTRIBUTION SCHEME BASED ON KEY UPDATING

ZHU Ling-Zhi[1], HE Rui[1] and ZHANG Jun-Ling[2*]

[1]Department of Computer and Information Science, Hunan Institute of Technology, Hengyang, 421002, Hunan, China

[2]Department of Cities and Tourism, Hengyang Normal University, Hengyang 421002, Hunan, China

Emails: lingzhi0825@163.com, 932900931@qq.com*

Submitted: Jan. 21, 2016 ***Accepted: Apr. 10, 2016*** ***Published: June 1, 2016***

Abstract-A random key pre-distribution scheme based on key updating (RKKU) was proposed, which is effective in wireless sensor networks. Firstly, the base station will randomly distribute some keys, a hash function and some code slices to each node. Furthermore, the RKKU scheme compares with the information of some random key to find the same key, and computes the communication key between two sensor nodes with one-way hash function. Since the one-way hash function can ensure that the attacker cannot use the obtained communication key to decipher the source key, it affects only two nodes communicate with each other. To assure the communication security, the key updating was designed based on code segment. The analysis shows that the proposed scheme can meet the security requirement of key management, and it also has less computation cost and storage cost than the existing schemes.

Index terms*: wireless sensor networks; code slices; pre-distribution; key updating, one-way hash function.

I. INTRODUCTION

With the technology development of sensors, low-power electronics and radio frequency identification (RFID), wireless sensor networks (WSNs) have become one of the most active frontier fields related to interdisciplinary studies and advanced technology. The researches on WSNs have tremendous significance and important scientific value. The sensor nodes just have limited resources and they are often deployed in harsh or adversary environments. Therefore, the nodes are easily to be captured by attackers. The security problems of wireless sensor networks are widely concerned.

Existing work has common problems, such as the isolation of sensor nodes, the waste in key pre-distribution, the leak problem of pre-shared keys, the ignorance or inability of key updating. Aiming to solve the hidden troubles of current key updating technologies for wireless sensor networks, key pre-distribution schemes are usually considered as the most viable solution. Nowadays, there are two main key pre-distribution schemes:

(1) All nodes communicate with the same pre-distributed key. It is maneuverable and easy to implement that any node will use the same pre-distributed key to communicate in WSNs. As the saying goes, each has both advantages and disadvantages. The disadvantages of this scheme are that the security performance is pretty low. Leakage of the master key may make the network facing huge threats.

(2) Using different key between any pair of nodes. Given this wireless sensor network has N nodes, if every node is allocated with N-1 different nodes in advance, there will be N*(N-1)/2 pre-distributed nodes in the network in total. Theoretically, the network security performance may be the best. In fact, it is a great waste of storage space that those resource limited nodes were allocated so much key in advance, and allocating so much key will increase computation and communication cost in WSNs.

It is inadequate to the mentioned two schemes for real sensor networks. Furthermore Eschenauer and Gligor introduced random graphic theory and put forward Random Key Pre-distributed Scheme [1] (E-G scheme). By comparison, The E-G scheme is easier to implement its algorithm and performs better in computation complexity and augment ability. But this scheme has some problems in practical applications, such as bad augment ability, ignoring or being

unable to update the key. How to effectively update the key and guarantee the subsequent security of networks are urgent problems in wireless sensor networks.

II. BASIC THEORY AND ASSUMPTION

a. System Assumption

Provided nodes are randomly distributed and only common nodes take part in the communication process, the fully distributed network model is adopted as system model. And then, Assumptions on system model are as follows:

(1) Every node has its exclusive mark;

(2) All nodes use the same configuration with equal computation, storage, communication resources;

(3) Nodes will hold still after deployed, no movement, unless captured or damaged.

Suppose attackers need obtain the message to spend T time from the captured nodes. What's more, during this time, the achievement of each two adjacent nodes are as follows:

(1) Exchanging information carried by code segments;

(2) Compounding programs according to combination and sequence of code segments;

(3) Entering current key to acquire the next key. Due to little number of code segment (k code segment) saved by each node, the time cost of exchanging code segment is relatively low. Meanwhile, getting new key by inputting current key needs only k mathematical operation which costs little time, thus the assumptions mentioned above are practical.

b. Network Model

Ordinary sensor nodes (called nodes or sensor nodes): aside from base station, the hardware configuration of all nodes is the same in wireless sensor networks. Every node should be responsible for completing basic task of the network (monitoring or locating person), giving preliminary process to acquired external data and transmitting the data to base station. Meanwhile every node shall undertake the task of transmitting or rebroadcasting the message sent by neighbor nodes or the entire network. The latter function is similar with the function of gateway or router.

User: Legal user or deplorer in WSNs. provided user has the same management authority and ability to access data with base station. Furthermore, user can manage WSNs, obtain data and publish related orders (update order) by the base station without energy consumption limits.

Network Deployment and Scale: In the practical application of WSNs, nodes are randomly shed into expected area by aircraft. This means that users or base station can not master and deploy position of nodes. In order to decrease the number of independent nodes (nodes without adjacent node, independent nodes form single node network), we assume that the dense of nodes in deployed area is sufficient enough.

c. Code Segment

Code segment pool should be set so large that the probability of getting two absolute same code segments close to zero when k code segment(s) were taken out from the pool any two times. In other words, the code segments saved by any two sensors are not exactly the same. New updating program which will be formed by the combination of some code segments inputs current communication key to execute the updating program. And then the output will be the new source key among nodes.

d. One-way hash function

In order to increase the security performance of key management scheme, One-way hash function is imported to improve the ability of this scheme to resist node captured. One-way hash function is the function which has the capacity to map a data of any length into a data of fixed length. Provided H is a Hash function, h is the given Hash value, x represent a data of any length, and then there will be an equation

$$h = H(x) \tag{1}$$

The supposed nature of a safe One-way hash function is as follows:

(1) One-way. For a given x, it is very easy to calculate h by the function $H(x)$. On the contrary, x can not be calculated by the function $H(x)$ within the regular time.

(2) For the given x, *we* find another information y to make *H(x) equal to H(y)*. However, the information y can not be calculated within the regular time.

(3) Strong collision resistance. We assume that there exists two information x and y which can make $H(x)$ equal to $H(y)$.But the information x and y can not be acquired by calculating within the regular time.

III. THE RKKU SCHEME

a. Random Pre-distribution

Like E-G scheme, a certain number of key materials will randomly be assigned to each node before the deployment of network. Firstly, the central sever will generate a code segment pool along with a key pool. Secondly, the server will randomly allocate some key material to every node, such as m keys ($K_{1.....}K_m$ e.g.), m keys' message ($MID_1...MID_m$ e.g.), k different code segment ($code\_segment_1...code\_segment_k$), the only node ID of the entire network, a hash function H (SHA-1 or RC5 e.g.) and a timer and so on. After key materials are distributed, all nodes will randomly be deployed in planned area.

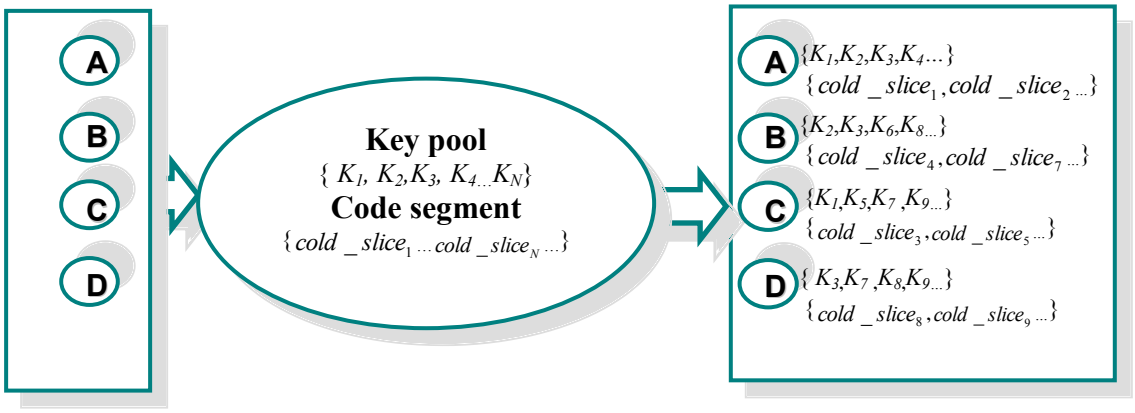

Figure 1 Random pre-distribution

b. Importing One-way hash function to Establish Direct Key

After the deployment of network, nodes will begin to find neighbor nodes by broadcasting pre-distributed keys' *ID* information of themselves and accepting the keys' *ID* information of neighbor nodes to check if there exists same keys' *ID* information in neighbor nodes' key *ID* information list. If exists, nodes will process all same keys' *ID* information through bitwise exclusive or operation. Given a and b were mutual neighbor nodes, then

$$K_{MID_{ab}} = K_{MID_1} \oplus K_{MID_2} \oplus ... \oplus K_{MID_s} \tag{2}$$

Where $\oplus$ represents the XOR operation, s indicates the number of keys with the same ID information. After operation, we encrypt the operating result $K_{\mathrm{MID\,ab}}$ with k code segments to broadcast to neighbor nodes. Once the code segments received, neighbor nodes will assemble the

acquired *k* code segments and their own *k* code segments according to the sequence regulated by r_k to form a program *p* which can be used to update keys. And then nodes use *s* identical keys to perform the XOR operation as follows:

$$K_{ab}^{0} = K_{a_1} \oplus K_{a_2} \oplus ... \oplus K_{a_s} \tag{3}$$

Next node *a* and node *b* will execute their first key update, inputting K_{ab}^{0} to execute update program *p,* so

$$K_{ab}^{1} = p\,(K_{ab}^{0}) \tag{4}$$

Meanwhile variable *i* execute a self-increase to record the number of updates. For an arbitrary key update (*i*+1 time), if their current source key is K_{ab}^{i}, after key update, the source key will be:

$$K_{ab}^{i+1} = p\,(K_{ab}^{i}) \tag{5}$$

Later on, executing hash operation on K_{ab}^{i+1}, there will be a new symmetric key:

$$h_{ab}^{i+1} = H(i+1, K_{ab}^{i+1}) \tag{6}$$

Node *a and* b will use the new symmetric key for encryption to execute normal communication.

c. Adjusting Dynamic Communication Radius as Support to Create Indirect Key

If two neighbor nodes fail to discover the same key *ID* information with those they have possessed after neighbor discovery that means symmetric key can not be established between the two nodes. RKKU and E-G [1] scheme adopt similar method to create indirect key by establishing a safe path with the help of third party node and the path must be safe and reliable. During the establishment of symmetric key, frequent negotiation and mutual message communication is needed among the nodes. Once the path was monitored, indirect key would be revealed.

This situation may exist in practical application: once a safe and reliable path could not be found between two neighbor nodes, the path could not be relied on to create keys either. Chen[2] proposed a new method in view of the above mentioned situation that not exist safe path: enlarging the communication radius among nodes of those unreal safe path adequately, in this way, further more, success rate of linking some special areas can be improved by creating indirect key and higher linking security can be granted to WSNs.

Table 1 represent the changes of the number of neighbor nodes which possesses a node with 30 original neighbor nodes that unable to establish direct key or seek a safe path to create indirect key in an special area of WSNs.

Table 1 the relationship between Communication radius and the number of neighbor nodes

Communication radius（multiple）	0	1	2	3
the number of neighbor nodes	30	120	270	480

It can be inferred form table 1 that enlarged communication radius of nodes increased the number of their neighbor nodes, if an shared key can be found among new nodes, that means a safe path exists, then indirect keys can be created via the path and communication radius of nodes are set as initial value at last.

Provided the density of nodes deployed in the network is large enough to ignore the necessity to ensure every node connect with all their neighbor nodes and guarantee information with very high probability at the same time, it would be acceptable that partial or very few mutual neighbor nodes did not establish safe communication links.

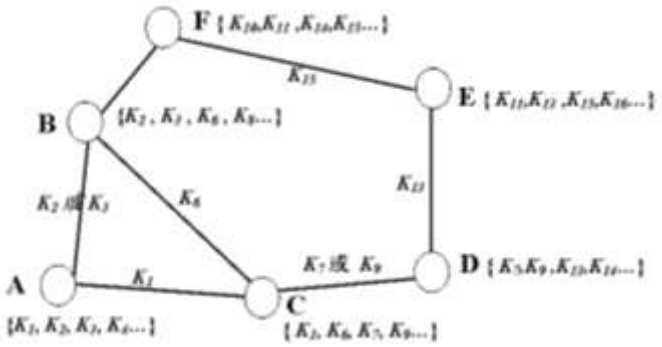

Figure 2 Establishing indirect key

d. Key Update

The timer will be triggered after WSNs nodes are deployed, when time exceeds ΔT the nodes will delete their pre-distributed keys saved by themselves and redundant keys and their hash value. Keys' corresponding *ID* information will be reserved (for adding new nodes) as well as source keys after update. What's more, some requirement should be met before executing key update that the hash value can be updated when the nodes are going to deliver data packets to their neighbors except for when time pause ΔT must longer than a certain cycle T. The

requirement not only can avoid excessive key update among nodes, but also be convenient for keeping the synchronism of key update.

Each node will use a variable *i* to record the number of time of keys' update. When node *b* received data packets from node *a*, firstly the value of variable *i* saved by node b will be compared with the value of variable *i* in the data packet send by node *a* and judge if the values of variable *i* are equal, if not, node *b* will input the key saved by itself which used between node a and bode b to execute key update and get a new key. Then executing hash operation to the new key, the communication key will be the obtained hash value which can be used to decipher the data packet send by node a. If variable *i* of node b and variable *i* in the data packet sent by node a are equal, so node b will be able to use the hash value saved by it self which used between node a and node b to decipher data packet.

e. Adding New Nodes

To some WSNs with special tasks, they may need longer working life. But the energy restriction of sensor nodes themselves makes them confront with troubles that sensors does not work due to energy shortage after a period of time. Furthermore, we need to add some nodes properly to network deployment area to maintain certain density of nodes. In this case, we need to make use of base stations, trustworthy third party or the negotiation among nodes to detect the validity of new nodes in order to resist attacks such as forgery and simulation.

Firstly, new nodes will send their key materials like *m* keys' *ID* information, *k* different code segments and the *ID* of the only node of the entire network to base station or trustworthy third party.

Later on, base station or trusted third party will operate related authentication. After confirming the validity of new nodes, base station or trusted third party will broadcast a message such as new nodes' *ID*. The node which receives the message will add new nodes' *ID* to its neighbor nodes' *ID* list.

Finally, if new nodes received message responded by base station or trusted third party, then executing neighbor discovery which means broadcasting new nodes' own pre-distributed keys' *ID* information and the information of new nodes themselves to check if there exists a pair of same *ID* by checking the list of neighbor nodes' key *ID* information and their own *ID*, if exists, then establishing symmetric key directly, if not, finding a safe path to establish symmetric key.

After realizing normal communication the nodes will delete their initial key and some redundant key or hash value to ensure communication security of most nodes of the network.

f. Removing Ineffective Nodes

As most of the deployment area of nodes is located in complex environments with remote location or harsh natural conditions, hardware restrictions of nodes themselves and limited battery power increased the probability of failure of nodes. How to delete ineffective nodes effectively is one of the important problems we need concern. In RKKU, due to the independence of each pair of nodes which have no relationship with other nodes, therefore when nodes exhaust all their energy, base station will broadcast the *ID* of nodes that used up their energy, neighbor nodes of which do not communicate with them so that the node get separated with the network. As for the condition that nodes were captured, we need suppose that an Intrusion Detection System is deployed in a base station or a network to analyze information and data of the network., when nodes exist abnormal or suspicious behavior and surrounding nodes make a agreement that the node is captured indeed and exists suspicious behavior, then surrounding nodes will interrupt communication with the node and report *ID* of the node to base station. Base station will broadcast *ID* of the node to other safe nodes of the entire network immediately after receiving the message and send key update order to all nodes of the network. After receiving key update order, the ineffective node will be input current source key to execute update program *P,* then making use of the output to perform hash operation, a hash will be gained as the latest key and the key update will be finished.

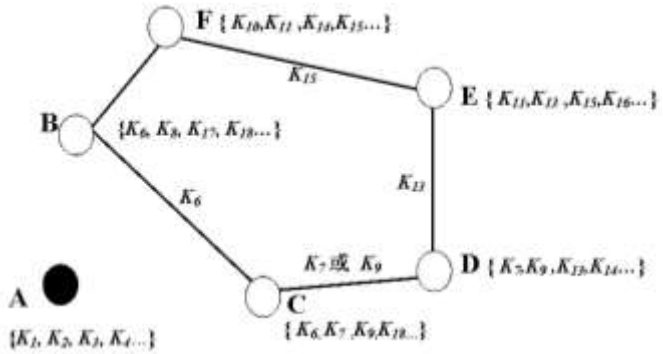

Figure 3 the event that node a is captured will trigger key update

V. PERFORMANCE ANALYSIS

It is important assessment standards to security performance and energy expense of nodes for the security performance of key management in WSNs. This scheme is mainly aimed at those attack models like forgery, monitoring and capture. Forgery is the process that attackers imitating the sender and communicating with the receiver directly and acquiring communication information from the receiver through forging messages. Attackers are monitoring communication among nodes and knowing communication details among nodes by receiving messages sent by nodes. Nodes capture is the way that attackers stealing information of nodes through physical nodes capture.

a. Nodes Connectivity Analysis

If symmetric keys can be established between mutual neighbor nodes, we define the link between two mutual neighbor nodes as the link that is capable of establishing symmetric key directly. In WSN, we can name the ratio occupied by the number of links that is capable of establishing symmetric key directly in total link number as safety link probability.

The way how RKKU deploy its nodes is basically similar with E-G scheme, according to Random Graph Theory, it can be referred that if random graph $G(N, p)$ with N nodes expect to achieve the given probability P_r of general connectivity, average degree d of nodes should meet the following condition at least:

$$d = (N-1)/N * [\ln(N) - \ln(-\ln(P_r))] \quad (7)$$

It can be referred from figure 4 that the increments of nodes is few as P_r increased, the bigger the N is, the smoother the curve will be and the expansion of network scale did not have much effect on nodes degree. Therefore the connectivity of the entire network can be ensured if nodes are able to communicate with their neighbor nodes with certain probability.

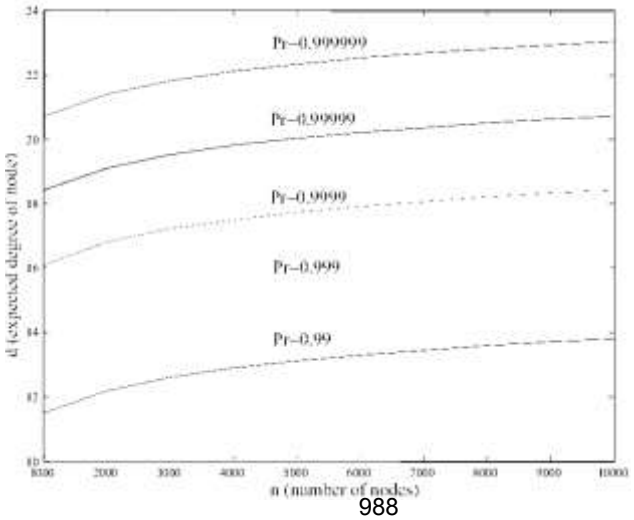

Figure 4 the relation curve of d, P_r and n [1]

According to formula (7) and [1], safety link probability p will be:

$$P=1-\frac{((P-K)!)^2}{(P-2K)!P!} \tag{8}$$

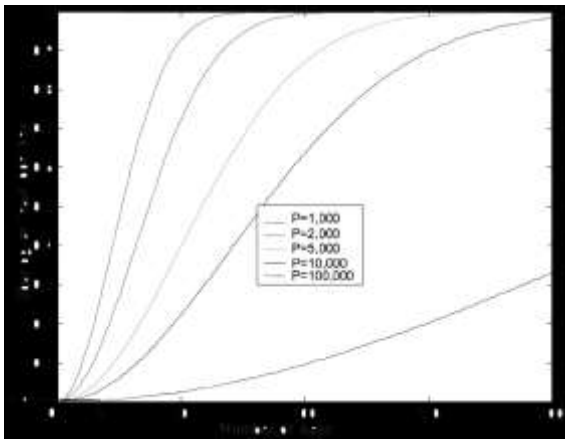

Figure 5 the relation curve of k, P and at least one shared key [1]

It can be seen from figure 5 that when the volume of key pool is 10000, just 75 pre-distributed keys will enable the probability that each two nodes have at least one shared key between them achieve 50%. If allocating 200 keys in advance the possibility of secure connection will achieve 98%. Increasing the value of K (when P value is fixed) or decreasing the value of P (when the value of K is fixed) can improve the possibility of secure connection of the network. Chan[3] has demonstrated the principle that when the density of nodes of sensor network is deployed at general level, once the probability of secure connection achieved 0.33%, shared session key can be established with no more than three hops between two mutual neighbor nodes and the goal can be achieved that the possibility of secure connection approaching 100%.

b. Security Analysis

As nodes usually are deployed in depopulated zone or a hostile environment, restrictions on hardware and energy make nodes vulnerable or easy to be captured, how to apply nodes safely is the first priority we need to concern. In order to obtain communication keys among nodes,

attacker may use such ways like forgery, monitoring, capture and so on. In this chapter we will analyze the security performance of the key update scheme proposed in the paper respectively according to the three mention means.

If attackers attack nodes through forgery, they may capture a node and extract prepared key information, next attackers will forge the information of keys or imitate valid nodes carry out active assaults to the rest nodes of the network so as to destroy trust relationships, forge sensitive information or spread attack. In RKKU, different nodes are required to share different communication keys, but the keys originate from initial keys distributed by the key pool. Initial key experienced code segments update and hash function operation. Moreover, communication process of code segments exchange has been encrypted, so attacking nodes via forge can be difficult.

If attackers choose monitoring as the mean to attack nodes, even if attackers are able to all communications of the network, due to the reason that all communications of nodes are encrypted by hash value, attackers still have no ability to monitor any communication among nodes. Even attackers get current hash value, *i.e.* communication key by other else complex methods, due to the characteristics of safe hash function, computing source key can be very difficult for attackers. Before next time pulse approaching, precious communication keys became invalid after update, as there is no communication agreement between new keys and previous keys, the security of the network will be guaranteed.

When attacker choose nodes capture as the mean to attack nodes, at worst attackers will completely capture nodes after a period of time t (obviously $t > \Delta T$) which means that attackers acquire all information of nodes including code segments allocated to captured nodes and corresponding hash values *i.e.* communication keys used among captured nodes and their free neighbor nodes. However, those information can not be applied to obtain communication keys among captured nodes which means the bad effect of nodes capture on security will be minimized because of the reason that initial keys randomly distributed by key pool has been deleted. The differences among diverse code segments and the characteristics of unilateral make the source password whose update pulse is ΔT can hardly inferred by hash values hence the influence on other else non-capture node can nearly be ignored. What's more, the more nodes captured by attackers, the more code segments will reap. With the help of these code segments, attackers may obtain all else code segments of some nodes. However attackers do not know the

initial keys which are randomly allocated in advance, let alone the source key used during above process. Attackers can not make use of the program built by obtained code segments to gain update keys of nodes whose all code segments are decoded. The scheme hence can guarantee security when attackers capturing nodes.

Figure 6 and Figure 7 show the communication link between the damage and the number of the captured nodes, when the total number of nodes N is 10000 in the wireless sensor networks. Among them, the program parameters p is 1, and k (the number of code fragments) is 50.

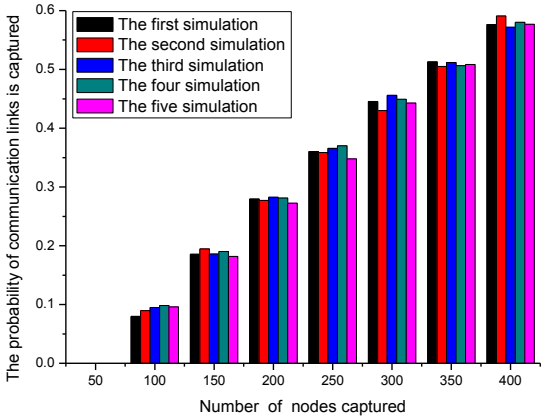

Figure 6 the number of captured nodes and the possibility of capturing communication links of non-captured nodes

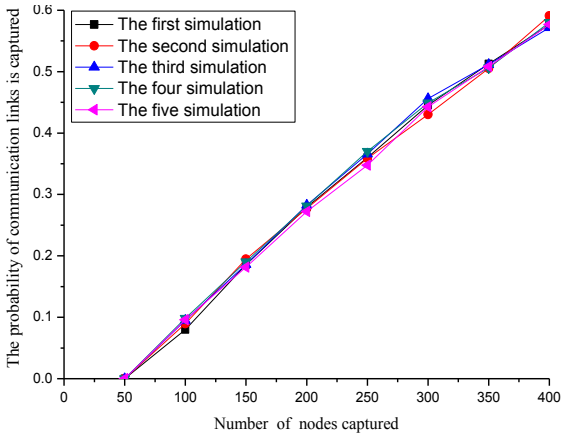

Figure 7 the number of captured nodes and the possibility of capturing communication links of non-captured nodes

Figure 8 also shows several classic key management schemes to be captured with impaired communication link between the numbers of nodes. In Figure 8, we can see that, as the total

number of nodes N is 10000, several classic key management schemes to be captured with impaired communication link between the numbers of nodes. In the E-G scheme, program parameters of p is 0.33 (two adjacent nodes can take up to three hops to achieve a secure link connectivity probability approaching 100%), and K is 200 (the number of key rings).The q-Composite scheme sets program parameters $q = 2$ and $p = 0.33$, The program parameters of the CSKU Scheme are $p = 1$ and $k = 50$ (the number of code fragments).

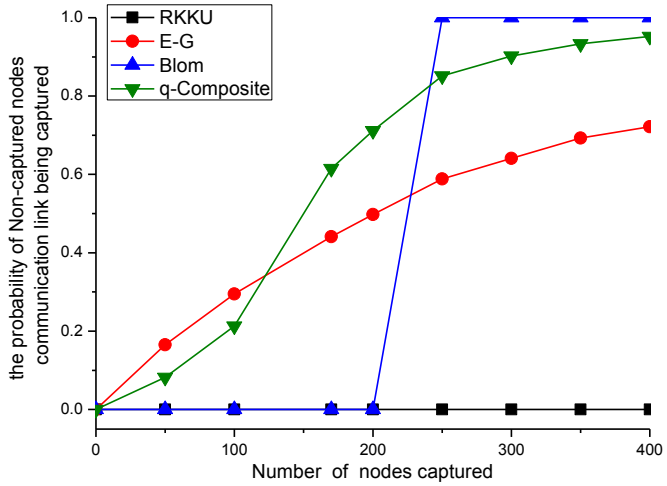

Figure 8 the number of captured nodes and the possibility of capturing communication links of non-captured nodes

c. Energy Expense Analysis

Reducing energy consumption and improving life network cycle are key points of WSNs key management. Generally WSNs completes tasks as a whole, but energy consumption of nodes directly influenced network life cycle. Therefore when energy expense should be fully considered when study WSNs key management scheme. In RKKU, my can discuss the problem according to computing, and storage.

c.i Communication Expense

AKYILDIZIF [4-8] has demonstrated that executing calculation is more energy saving than transmitting messages for sensors. The electrical energy consumption of executing 3000 computations equal to that of transmitting one bit information, the key management scheme hence can add operations properly so as to decrease data transmission. Dynamic key management scheme which based on reliable key distribution center and frequent key update became

impractical for transmitting too much data information. In order to simplified the comparison we assume that all schemes the same encryption algorithm to generate keys and the length of information to be encrypted are equal. The E-G scheme and its modified version need not transmit data in large quantity and too much communication expenses, but long tern usage of symmetric keys without update has brought new hidden troubles [9-13]. In RKKU, communication expenses of nodes mainly generated from exchange code segments when exchanging key materials. When keys are at the stage of update, nodes need only transmit the values of variable i and cost very few expenses. Table 2 presented communication expenses of several common key management schemes and compared them with expenses of key update.

Table 2 the comparison between communication expenses of key management schemes and expenses of key update

Key management scheme	beforehand share	E-G	Blom	RKKU
Communication expenses	1	$(1+d)*k$	$(\lambda+1)*(d+1)$	$k*d$
Key update expenses	/	/	/	pass value of i

c.ii Storage and Compute Expenses

Provided a single instruction character occupied a bit(s) storage space, the amount of single operation occupied b bit(s), pre-distributed keys occupied c, hash function occupied f bit(s), the average scale of symmetric key g bit(s), the average neighbor number of nodes was d, thus the storage space required by RKKU was s bit(s) which met the equation that $s = k*(a+b)+g(d+1)+f$,as $a+b << g$, $f << g$,pre-distributed and invalid keys would be deleted after a pulse represented by ΔT while sensor network finished its deployment, so the storage space needed by RKKU approximate s bit(s) which met the equation that $s \approx g*(d+1)$.

RKKU requires very few expenses, key update need only executing hash operation once and arithmetical operation k times, few expenses indeed. Table 3 presented the comparison made among several common key management schemes' storage expenses.

Table 3 Comparison made among several common key management Scheme s' storage expenses

Key management schemes	beforehand share	E-G	Blom	RKKU
Storage expenses	1	$2k$	$(\lambda+1)*2$	$g*(d+1)$

VI. CONCLUSIONS

In this paper, the writer discusses the advantages and disadvantages of several classic schemes and the modified schemes, and analyzes the problem. With the key ID information, the RKKU scheme discovers neighbor nodes and saves key ID information corresponding to the key of the session up, and then the introduction of a one-way hash functions and code segment updated technology to calculate key communications between nodes. The RKKU scheme can achieve low power consumption and secure communications, between the base station and nodes, nodes and node efficient. Compared with the classic schemes, The RKKU scheme firstly stores only part keys and pre-assigned key ID information, and reduces the storage space. Although the introduction of a one-way hash functions and code segment, its storage space is very small. Secondly, the introduction of one-way hash function ensures that even communication key compromise, the attacker cannot use the obtained communication key to decipher the source key, so that it affects only two nodes communicate with each other, there is no impact on other nodes. Finally, the updating technologies based on the code segment make wireless sensor networks with minimal energy cost to complete the update node key, even if the communication key compromise, after the next moment comes, the new communications key will automatically overwrite leaked key .With regard to our future work, we plan to design a distributed dynamic updating key management algorithm and make the RKKU scheme a totally distributed dynamic scheme.

AKNOWLEDGEMENT

This work was financially supported by Hunan Province College Research Project of the Teaching Reform (Xiang Jiao Tong [2015]291-562), The Project Development Plan of Science and Technology

of Hengyang City (No. 2014KG63, 2014KG38), the scientific research project of Hunan Institute of Technology(HY13001).

REFERENCES

[1] Eschenauer, L., & Gligor, V. D, "A Key-Management Scheme for Distributed Sensor Networks", Proceedings of the 9th ACM conference on Computer and communications security, 2002, pp.41-47.

[2] Chen G, Wang G J, "Random key pre-distribution scheme based on secure connectivity detection", Computer Engineering & Applications, Vol.45, No.6, 2009, pp.109-111.

[3] Chan H, Perrig A, & Song D, "Random key pre-distribution schemes for sensor networks", Security and Privacy, Proceedings of 2003 Symposium on IEEE, Vol.98, 2003, pp.197-213.

[4] AKYILDIZIF，SUW，SANKARA SUBRAMAN IAM Y，et al. "A survey on sensor networks", IEEE Communications Magazine, Vol.40, No.8, 2002, pp. 102-114.

[5] Chan H W， Adrian P，Song ` D. "Random Key Pre-distribution Scheme for Sensor Networks", 2003 IEEE Symposium on Research in Security and Privacy,2003, pp. 197-213.

[6] Huang D, Mehta M, Van d L A, et al. "Modeling Pairwise Key Establishment for Random Key Predistribution in Large-Scale Sensor Networks", IEEE/ACM Transactions on Networking, Vol.15, No.5, 2007, pp. 1204-1215.

[7] Blom R. "An Optimal Class of Symmetric Key Generation Systems", Lecture Notes in Computer Science, Vol.209, No.2, 1984, pp. 335-338.

[8] Du W, Deng J, Han Y S, et al. "A Pair Wise Key Pre-Distribution Scheme for Wireless Sensor Networks", International Journal of Innovative Technology & Exploring Engineering, Vol.8, No.2, 2003, pp. 42-51.

[9] Blundo C, Santis A D, Herzberg A, et al. Perfectly-Secure Key Distribution for Dynamic Conferences [J]. Information & Computation, Vol.146, No.1, 1996, pp. 1-23.

[10] Liu D, Ning P, Li R. "Establishing pairwise keys in distributed sensor networks", ACM Transactions on Information & System Security, Vol.8, No.1, 2005, pp. 41-77.

[11] Younis M, Ghumman K, Eltoweissy M. “Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Networks”, IEEE Transactions on Parallel & Distributed Systems, Vol.17, No.8, 2006, pp. 865-882.

[12] Eltoweissy M, Moharrum M, Mukkamala R. “Dynamic key management in sensor networks”. IEEE Communications Magazine, Vol.44, No.4, 2006, pp. 122-130.

[13] Dolev D, Yao A C. “On the Security of Public Key Protocol”, IEEE Transactions on Information Theory, Vol.29, No.2, 1983, pp. 198-208.

[14] Xu J, Qian H, Ying W, et al. “A deployment algorithm for mobile wireless sensor networks based on the electrostatic field theory”, The International Journal on Smart Sensing and Intelligent Systems, Vol.8, No.1, 2015, pp. 516-537.

[15] Bai, Q., & Jin, C, “Image fusion and recognition based on compressed sensing theory”, International Journal on Smart Sensing & Intelligent Systems, Vol.8, No.1, 2015, pp. 159-180.

[16] Qiao J, Liu S, Qi X, et al, “Transmission power control in wireless sensor networks under the minimum connected average node degree constraint”, The International Journal on Smart Sensing and Intelligent Systems, Vol.8, No. 1, 2015, pp.801-821.