ICIC Express Letters
Volume **11**, Number **2**, February **2017**

 ISSN 1881-803X
pp. **365–372**

LOCATION-BASED RANDOM KEY PREDISTRIBUTION SCHEME OF WIRELESS SENSOR NETWORKS

Wenju Liu, Yuhui Jia and Ze Wang

School of Computer Science and Software
Tianjin Polytechnic University
No. 399, Binshui West Road, Xiqing Dist., Tianjin 300387, P. R. China
jyhjyh188@sina.com

Received July 2016; accepted October 2016

Abstract. *For connectivity and security problems of wireless sensor networks (WSN), a location-based random key pre-distribution scheme is proposed in this paper. The scheme utilizes location information of nodes in WSN to establish corresponding relationship between location and key pool. Higher connectivity rate can be achieved under the conventional key storage through key deletion strategy based on node location distribution and the increase in the probability of sharing same keys between adjacent nodes. Theoretical analysis and simulation data show that this scheme improves various performance of WSN on the basis of security enhancement, such as communication overhead and resiliency performance.*
Keywords: Wireless sensor networks, Key pre-distribution, Location-based key correlation, Key deletion strategy

1. **Introduction.** In recent years, wireless sensor networks (WSN) have been widely applied in such fields as medical treatment, military monitoring and environment monitoring. WSN are formed by large quantities of sensor nodes [1-3] with low cost, low power dissipation and such functions of information collection, data processing and wireless communication through self-organization. These sensor resources are relatively limited, with low cost and small volume. Security problem of wireless sensor networks has become a significant problem [4-6]. Key management is the most fundamental and important part in security mechanism. Thus, it is very necessary to introduce key management mechanism in order to enhance network security and anti-attacking property [7-10].

Current researches indicate that pre-distribution scheme is the most appropriate key management scheme of WSN. For key pre-distribution problem, there are all kinds of solutions. For example, key pre-distribution scheme based on symmetrical and balanced incomplete blocks is established in [11]. [12] proposes a key pre-distribution scheme based on location information. [13] puts forward a key pre-distribution scheme based on hexagon deployment.

Eschenauer et al. first came up with the random key pre-distribution scheme (E-G scheme). The scheme is divided into three stages: key pre-distribution stage, key discovery stage and key path establishment stage. This scheme has large network size, but resiliency performance is very poor. Castelluccia and Spognardi proposed the RoK scheme [14]. This scheme enhances the robustness, but connectivity is significantly insufficient. Sarimurat and Levi put forward a key pre-distribution scheme based on Hash graph (HaG scheme) [15]. The key is generated by Hash algorithm, so higher security is achieved. This scheme has high connectivity, but communication load and computation overhead are relatively high. This paper is proposed based on location key correlation policy (LKC) scheme. The corresponding relationship between key and location is established through correlation with node location information. A large number of keys are

stored in key pre-distribution stage. After the nodes are deployed, few keys can reach high connectivity through key deletion strategy based on node location distribution, i.e., increase the probability that neighbor nodes share the same keys. Thus, the overhead is reduced and security is improved.

2. System Model.

This section mainly introduces system model establishment, including key pool generation, key ring establishment, and pairwise key establishment.

2.1. System overview and relevant parameters.

Sensor nodes have very limited amount of energy reserve that limits their lifetime to a small period of time. Thus, new sensor nodes need to be deployed to the network in some intervals. And new nodes will replace original nodes for communication.

Table 1 lists symbols involved in this paper and corresponding meanings.

Table 1. List of symbols used in our scheme

Symbol	Definition
P	Key pool size
G_w	Generation window
KP^i	Key pool at generation i
k_t^i	Key with index t at generation i
$h(\cdot)$	Secure hash function h: $\{0,1\}^* \rightarrow \{0,1\}^{160}$
$f(\cdot)$	Hash function h: $\{0,1\}^* \rightarrow \{0,1\}^{\frac{p}{g}}$
g	Number of keys in a key ring
n	Number of key rings in a node
m	Number of keys in a node
k	The ratio of the horizontal coordinate to the area length
$k^{'}$	The ratio of key ring ID and key pool size
x	Horizontal coordinate of the node
X	Length of area
w	Number of reserved key ring groups
v	Number of deleted key ring groups

For the convenience of description, *generation* is used to express time interval of nodes in the networks. The life boundary is expressed as Gw. A node deployed at generation i will drain its battery before generation $i + Gw$ and each generation period is assumed to be 1 in the rest of the paper.

2.2. Key pool generation.

Key pool is updated at each generation of our scheme. P random generated keys will be generated initially in key pool. When the generation period ends, two continuous keys will generate a secure hash function $h(\cdot)$, such as SHA-1.

Initial key pool is defined as follows:

$$KP^0 = \left\{k_1^0, k_2^0, k_3^0, k_4^0, k_5^0, \ldots, k_{p-1}^0, k_p^0\right\}$$

wherein, every value of k_i^0 is random, and the last key in each generation is the randomly generated key.

If the key of key pool at generation i is $KP^i = \left\{k_1^i, k_2^i, k_3^i, \ldots, k_{p-1}^i, k_p^i\right\}$, then the key of key pool at generation $i+1$ is $KP^{i+1} = \left\{k_1^{i+1}, k_2^{i+1}, k_3^{i+1}, \ldots, k_{p-1}^{i+1}, k_p^{i+1}\right\}$, wherein, $k_t^{i+1} = h\left(k_t^i \oplus k_{t+1}^i\right)$.

2.3. **Key ring establishment.** In the stage of key pre-distribution, the node randomly chooses g adjacent keys from the key pool as a key ring and stores n groups of key rings. Hence, the number of keys stored by the node is $m = n \times g$.

A disconnected pseudorandom function $f(\cdot)$ is applied to storing the key group generated by the node. Based on lifetime distribution of the observed node, it is better for the value of g to be close to $Gw/2$.

2.4. **Pairwise key establishment.** Nodes start pairwise key establishment phase right after being deployed to the environment. Nodes prestore at generation t of keys, and the keys are randomly deployed in the target area. When a sensor node A is deployed to the network at generation t, it broadcasts a message containing these values. Neighbor nodes can use this message to check whether they have at least one key in common. If neighbor nodes own the same keys, those two nodes should find a trusted (encrypted) path for communication. If there are no same keys, the node will wait for the generation $t+1$. At generation $t+1$, the key of generation t will generate the key of generation $t+1$ by Hash function according to the method of key pool generation. Once the key of generation $t+1$ is generated, the key of generation t will be deleted immediately to guarantee forward secrecy.

3. **LKC Scheme.** This section mainly introduces LKC scheme and carries out theoretical analysis and security analysis of this scheme. This scheme is mainly divided into 3 stages: key pre-distribution, key confirmation and key discovery.

3.1. **Key pre-distribution.** The node chooses n groups of keys from the key pool, and each group includes g keys. In other words, $m = n \times g$ different keys are stored. Besides, ID information of the first key in each group is stored. The ID stored is $ID = \{id_1, id_2, id_3, \ldots, id_{n-1}, id_n\}$.

3.2. **Key confirmation.** We suppose that in ideal status, the node will acquire its location information immediately after it is deployed, and this stage is executed at once. Key deletion strategy of LKC is as follows.

After node location is confirmed, the node will figure out the specific value of its horizontal coordinate (x) and regional length (X) $k = x/X$, and compares this specific value with the specific value of key ring ID and key pool size P $k^{'} = ID/P$. Some key rings close to k are retained and other key rings far away from k are deleted, as shown in Figure 1. We suppose the length of target area is AB, the node is deployed to C, and the size of key pool is P. The node will retain some keys whose ID value is close to k.

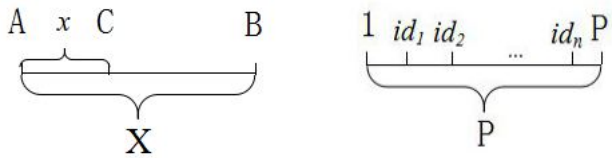

FIGURE 1. Key division diagram

The number of key rings retained is related to node connectivity. If the number of key rings retained is too small, the connectivity will be too low. If the number of key rings retained is too large, the waste will be caused. In the subsequent simulation experiment, this paper analyzes the relationship between the number of key rings retained and connectivity through examples.

After the node is deployed, w groups of keys will be retained. The number of key groups deleted is $v = n - w$. The types of keys retained involve 3 situations:

① When $k <= v/n$, the node will retain w groups of keys whose ID is small in the key pool;

② When $k > w/n$, the node will retain w groups of keys whose ID is large in the key pool;

③ When $k > v/n\delta\delta k <= w/n$, the node will retain the key groups whose ID is within $\left[(k^{'} \times n - \lfloor w/2 \rfloor)/n \times P, (k^{'} \times n + \lfloor w/2 \rfloor)/n \times P\right]$. There are w key ring groups in total, and other key rings are deleted. Finally, the node will retain $w \times g$ keys.

The detailed process is shown in Algorithm 1.

Algorithm 1 Key confirmation

```
k = x/X
if (k <= x/X) then
  for (id_1 : id_n) do
    ID = {id_1, id_2, id_3, ..., id_w}
  end for
else if (k > w/n) then
  for (id_1 : id_n) do
    ID = {id_w, id_{w+1}, id_{w+2}, ..., id_n}
  end for
else
  for (id_1 : id_n) do
    ID = {id | (k' × n − ⌊w/2⌋)/n × p <= id <= (k' × n + ⌊w/2⌋)/n × p}
  end for
end if
```

3.3. **Key discovery.** After the key is confirmed, the node will broadcast key ID to seek the key node which has the same key with it in the neighbor nodes. After the key node is found, a trusted path can be established at once for communication.

3.4. **Scheme analysis.** To narrate simply, we make communication radius dr of the node, the basic unit of length ($dr = 1$). Each node has m neighbor nodes on the average. In other words, there are $m + 1$ nodes in the communication area $\pi dr^2 = \pi$. If the target area is F, the total number of nodes is $N = \frac{F \times (m+1)}{\pi}$.

We suppose node u and node v are neighbor nodes, and $k = g \times n$ keys are stored in the pre-distribution stage. If the key is not deleted, the probability that u and v have the same key is $P = 1 - \left(1 - \frac{g \times n}{P}\right)^{g \times n}$. The node deletes some keys which are unrelated to location in key confirmation stage. Then, the probability that u and v have the same key is $P = 1 - \left(1 - \frac{k - g \times v}{P}\right)^{k - g \times v}$.

This paper mainly discusses the relationship between the key ring retained and connectivity. We suppose the size of key pool, total number of network nodes, average number of neighbor nodes and the number of keys retained are known. The relationship between the size of keys stored and local connectivity is shown in Figure 2.

3.5. **Security analysis.** Forward secrecy: after the node is deployed, the keys of each generation are different. After a new generation of keys is generated, the previous generation of keys will be deleted immediately, which guarantees forward secrecy.

Backward secrecy: after the node is deployed, its life is limited. If it is captured, limited key is gained. As times goes on, the key will lose use value, which guarantees backward secrecy.

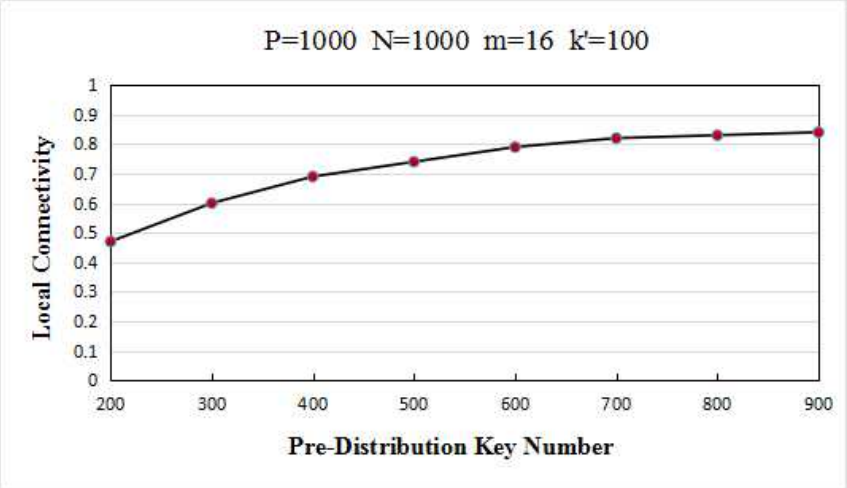

FIGURE 2. The probability that neighbor nodes have the same key

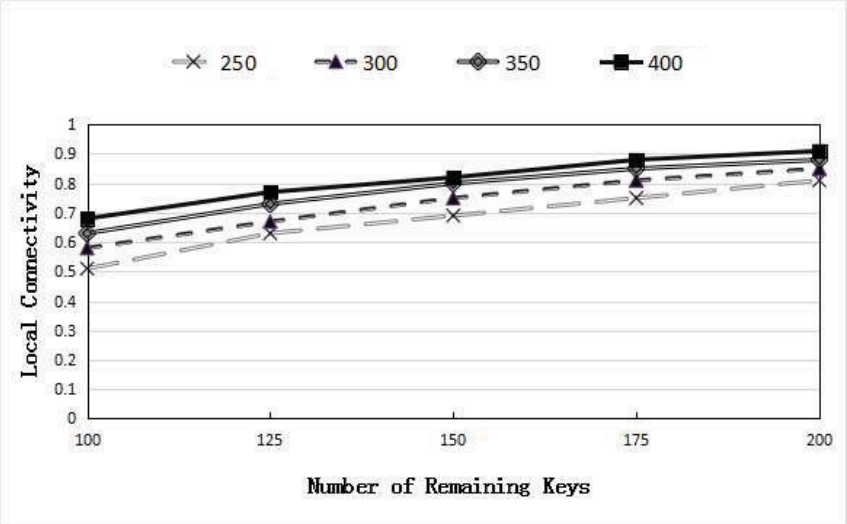

FIGURE 3. Local connectivity of LKC scheme

4. **Simulation Experiment.** RoK scheme has strong robustness, but the connectivity is insufficient. HaG scheme owns high connectivity and resiliency performance. However, the communication overhead and computation overhead increase greatly. LKC scheme prestores large quantities of keys and utilizes location information after node deployment to screen keys so as to increase the probability that neighbor nodes share the same keys.

4.1. **Simulation setup.** We preset the size of key pool to 10000 keys. In the 400m×400m area, 1000 sensor nodes are placed at random. Their communication radius is 40m. The lifetime of sensors (Gw) is set to 10, and sensor nodes have a random lifetime to normal distribution function with the mean value of $Gw/2$ and standard deviation of $Gw/6$. The value of g is set to $5 = Gw/2$. In addition, we have assumed that each generation consists of 10 small time units called *rounds*. At the beginning of every generation, the dead nodes will be replaced by new random nodes immediately. The mean value is taken after the simulation of 30 generations and operation for 25 times.

4.2. **Connectivity analysis.** Global connectivity: RoK, HaG and LKC schemes can reach 100% global connectivity.

Local connectivity: the possibility that any two adjacent nodes share the same key. Local connectivity of LKC scheme is shown in Figure 3. In Figure 3, there are 4 series

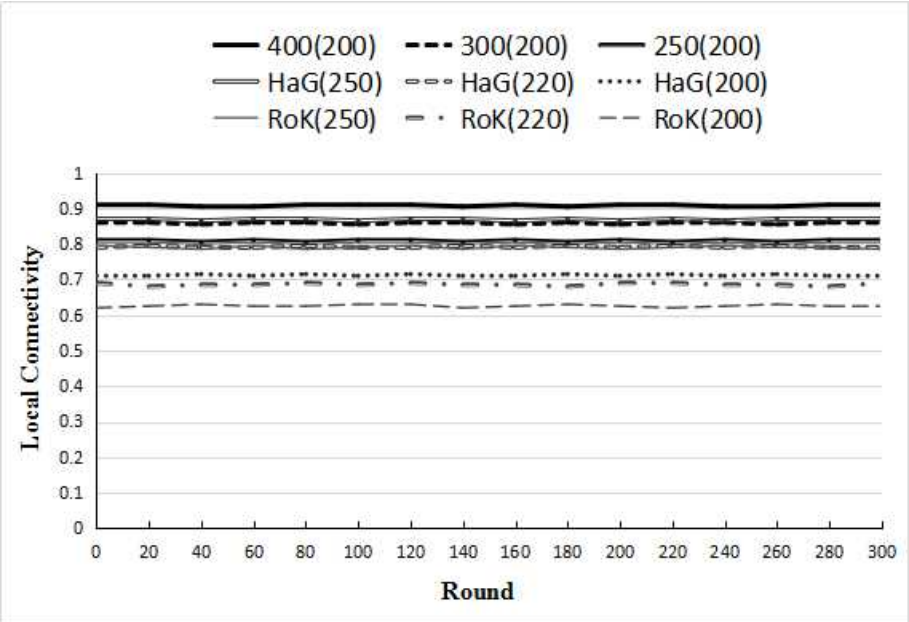

FIGURE 4. Local connectivity of RoK, HaG and LKC schemes

which represent 250 keys, 300 keys, 350 keys and 400 keys in total, respectively. The connectivity of 250 keys is the minimum, while the connectivity of 400 keys is the maximum. The horizontal coordinate represents the number of remaining keys, and the vertical coordinate represents the connectivity. As shown in the figure, when 200 keys from 250 keys are surplus, and 150 keys from 400 keys are surplus, the connectivity has reached 80%. Thus, how to choose a proper key also becomes an important link. For wireless sensor networks, 80% local connectivity is sufficient for secure communication.

Figure 4 shows connectivity comparison of RoK, HaG and LKC schemes. It is known from the figure that, LKC can achieve better connectivity than RoK and HaG with fewer keys. 400(200) means there are 400 keys in total and 200 keys are surplus. 300(200) means there are 300 keys in total and 200 keys are surplus.

4.3. **Resiliency analysis.** The attacker will capture 1, 3 and 5 nodes at random in per round, and obtain the keys stored in the memory by the node for attacking. In this paper, LKC scheme (300(175)), RoK scheme (250) and HaG (220) whose connectivity is close to 80% respectively are chosen for comparison. Under the same connectivity, the number of keys used in LKC scheme is fewer than that in RoK and HaG schemes. Thus, LKC has strong resiliency performance. The results are shown in Figure 5.

4.4. **Communication overhead and computation overhead.** After the node is deployed, it will communicate with the nodes nearby. If there is the same key, a secure path will be established. Figure 6 shows comparison of communication overhead of LKC with RoK(250) and HaG(220). It is known from the figure that LKC scheme in which some keys are deleted greatly reduces communication overhead, compared with RoK and HaG schemes.

LKC scheme increases the overhead of location percentage calculation after positioning, compared with RoK and HaG schemes. However, the computation overhead is acceptable, compared with Hash algorithm and reduced number of keys.

4.5. **Storage overhead.** Table 2 and Table 3 contrast memory use situations after and before deployment. To make storage overhead comparable, this paper chooses several data with the connectivity close to 80% for comparison. It is found that although LKC scheme needs to store a large number of data, memory use amount does not increase through

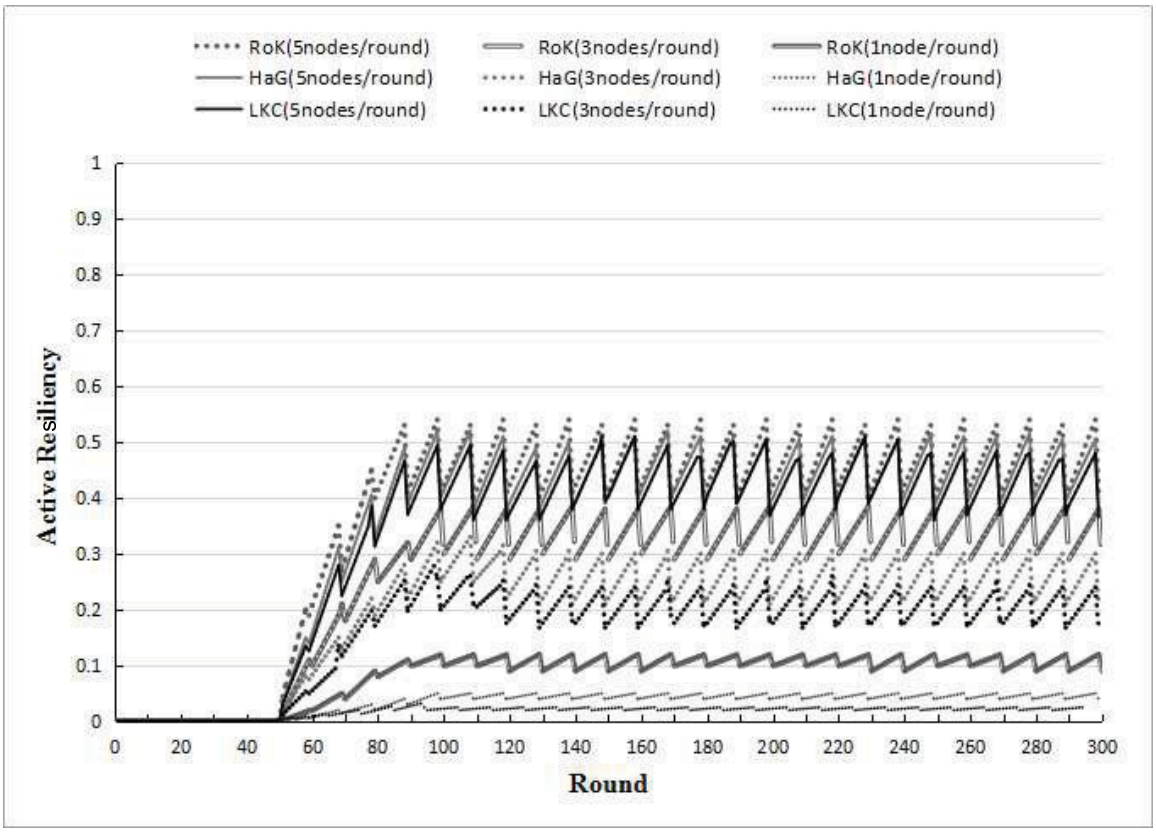

FIGURE 5. Active Resiliency of RoK, HaG and LKC schemes

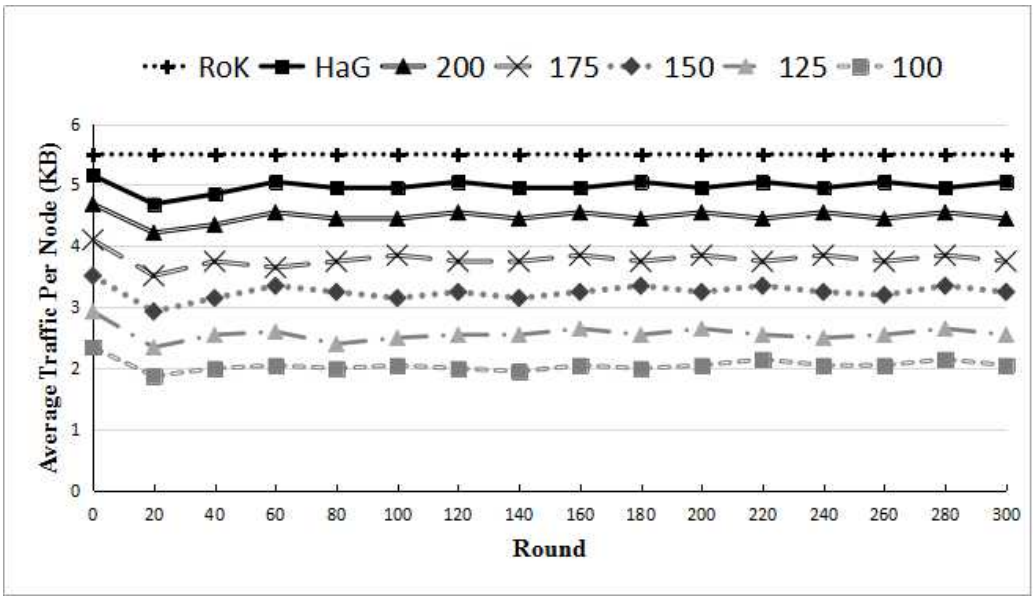

FIGURE 6. Communication overhead of RoK, HaG and LKC schemes

TABLE 2. Memory use situation before deployment

Method	Rok(250)	HaG(220)	400(150)	350(150)	300(175)	250(200)
RAM	5002	4402	8002	7002	6002	5002

discarding unnecessary keys after the node is deployed. On the contrary, to reach higher connectivity, the number of keys used decreases, and memory use amount also decreases.

Table 3. Memory use situation after deployment

Method	Rok(250)	HaG(220)	400(150)	350(150)	300(175)	250(200)
RAM	4002	3522	2402	2402	2802	3202

5. **Conclusion.** LKC scheme has higher connectivity and resiliency performance than RoK and HaG schemes. Although large quantities of keys are stored in the predistribution stage, memory use amount decreases instead after keys are deleted. LKC scheme achieves high connectivity with few keys through utilizing location information after the node is deployed, discarding some keys and increasing the probability that neighbor nodes share the same keys. It is found from the experiment that, with the same number of keys, LKC scheme owns higher connectivity and security as well as lower communication load. With the same connectivity, LKC scheme has higher resiliency performance.

In this paper, we propose a scheme based on location key correlation policy. The scheme utilizes location information of nodes in wireless sensor networks to increase the probability of sharing the same keys between adjacent nodes. In order to further improve the security performance of wireless sensor networks, the research on key pre-distribution scheme is still a direction which is worth exploring.

REFERENCES

[1] E. Khan, E. Gabidulin, B. Honary and H. Ahmed, Matrix-based memory efficient symmetric key generation and pre-distribution scheme for wireless sensor networks, *IET Wireless Sensor Systems*, vol.2, no.2, pp.108-114, 2012.

[2] Z. Su, C. Lin and F. Y. Ren, Hash chain based random keys pre-distribution scheme in wireless sensor networks, *Chinese Journal of Computers*, vol.1, pp.30-41, 2009.

[3] W. Bechkit, Y. Challal, A. Bouabdallah and V. Tarokh, A highly scalable key pre-distribution scheme for wireless sensor networks, *IEEE Trans. Wireless Communications*, pp.948-959, 2013.

[4] R. K. Kodali, Key management technique for WSNs, *Region 10 Symposium IEEE Kuala Lumpur*, pp.540-545, 2014.

[5] M. Ergun, A. Levi and E. Savas, A resilient key pre-distribution scheme for multiphase wireless sensor networks, *International Symposium on Computer and Information Sciences*, pp.375-380, 2009.

[6] Y. Sun, Y. Cao and L. Tang, A multi-phase key pre-distribution scheme based on hash chain, *The 9th International Conference on FSKD*, pp.2061-2064, 2012.

[7] W. Tong, J. Liang, X. Jin and Z. Li, A survey on key pre-distribution scheme of distributed WSNs, *Instrumentation Measurement Computer Communication and Control*, pp.242-246, 2013.

[8] M. Rezaeirad, M. Orooji, S. Mazloom, D. Perkins and M. Bayoumi, A novel clustering paradigm for key pre-distribution: Toward a better security in homogenous WSNs, *Consumer Communications and Networking Conference*, pp.308-316, 2013.

[9] T. Feng and J. F. Ma, A general key seed management and assignment model for wireless sensor networks and application, *Journal of Computer Research and Development*, vol.1, pp.146-153, 2008.

[10] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz and A. Khalili, A pairwise key pre-distribution scheme for wireless sensor networks, *ACM Trans. Information and System Security*, pp.228-258, 2005.

[11] G. M. Xia, Z. G. Huang and Z. Y. Wang, A key pre-distribution scheme for wireless sensor networks based on the symmetric balanced incomplete block design, *Journal of Computer Research and Development*, vol.1, pp.154-164, 2008.

[12] T. Kwon, J. Lee and J. Song, Location-based pairwise key predistribution for wireless sensor networks, *IEEE Trans. Wireless Communications*, vol.8, 2009.

[13] X. L. Yan and X. H. Ye, Random key predistribution scheme for sensor networks based on hexagon deployment model, *Application Research of Computers*, vol.4, pp.1457-1461, 2012.

[14] C. Castelluccia and A. Spognardi, RoK: A robust key pre-distribution protocol for multi-phase wireless sensor networks, *International Conference on Security and Privacy*, pp.351-360, 2007.

[15] S. Sarimurat and A. Levi, HaG: Hash graph based key predistribution scheme for multiphase wireless sensor networks, *International Conference on Communications*, pp.2079-2083, 2013.