A new key establishment method based on random key pre-distribution in sensor networks

Sujun Li, Boqing Zhou, Decheng Miao, Yun Cheng, Jie Wu, IEEE Fellow

Sensor networks are often deployed outdoors and are more susceptible to various attacks. In order to protect communication between nodes, scholars have proposed key management schemes. The most popular among them is the key pre-distribution scheme. This scheme has the following contradiction: in order to improve security, the number of keys pre-distributed to nodes should be minimized as much as possible. However, as the number of pre-distribution keys decreases, the network may no longer be securely connected, resulting in wastage of nodes. In this paper, we propose a new key establishment method to address this issue. Analysis and simulation show that the proposed scheme can reduce the number of pre-distribution keys in the original schemes while ensuring secure network connectivity.

Index Terms—*sensor networks, key predistribution, share key establishment*.

I. INTRODUCTION

Sensor networks are often deployed outdoors or even in hostile environments, making them vulnerable to various attacks [1]-[2]. In order to achieve secure communication between nodes, a shared key must be established between them. So far, relevant researchers have proposed various key pre-distribution schemes [3]-[11]. Eschenauer and Gligor proposed the first key pre-distribution scheme, named E-G scheme [3]. In the scheme, nodes randomly and nonrepetitively select some keys and their IDs from the key pool. If two nodes have common pre-distribution keys, they can use these common keys to establish a shared key. Based on the scheme, Chan et al. proposed a q scheme [4]. In this scheme, the prerequisite for establishing a shared key between two nodes is that there are no less than q shared pre-distribution keys between them. Kittur and Pais use Combinatorial Design to enable shared keys to be established between nodes within a cluster [5]. Maciej and Alexander proposed a key management scheme based on spanning trees [6]. Zhou et al. proposed two key management models for applications with large deployment errors [7] - [8]. In addition, Zhou et al. proposed a new key pre-distribution model [9]. This model can resist replication mobile node attacks and false data injection attacks. Msolli et al. proposed a new method for constructing a key pool [10], in which the key pool contains keys and their hashed values. Due to the adversary being unable to obtain the original key after obtaining its hashed key, its security performance is improved as compared to the E-G scheme [3]. In [11], Du et al. proposed a scheme for constructing a key pool using t-degree binary variable symmetric polynomials [12].

In key pre-distribution schemes, there is always a contradiction: to improve the scheme's resilience, the key information pre-distributed to nodes should be minimized as much as possible. However, as the number of pre-distribution keys decreases, some nodes in the network may become isolated, leading to node waste. Therefore, research should be conducted on methods for establishing shared keys to achieve the ability to establish shared keys between nodes even with a small number of keys distributed in advance.

In this paper, we propose a new scheme for shared keys establishment. Our main contributions are as follows:

1) In our scheme, the pre-distribution times of a t-degree binary variable symmetric polynomial will not exceed t, indicating that the shared key directly established between nodes has perfect resilience.

2) We propose a new method for establishing shared keys. This method includes the following three steps: the first step is to directly establish a shared key between two nodes; The second step involves a wider range of nodes participating in the establishment of path keys; The third step involves the establishment of path keys by the common neighbors of two nodes. In this method, although only a small number of keys pre-distributed to a node before deployment, the probability of establishing a shared key between two nodes after deployment is high.

The letter is organized as follows. the proposed scheme will be presented in Section II. The theoretical and experimental results will be described in Section III. At last, the conclusion will be made in Section IV.

II. OUR SCHEME

Here, we will introduce our scheme from the following two aspects: one is key pre-distribution, the other is shared keys establishment.

Corresponding authors: Boqing Zhou; Decheng Miao.

S. Li, B. Zhou and D. Miao are with the School of Information Engineering, Shaoguan University, Shaoguan 512005, China, E-mails: {lsj_paper, zbq_paper} @163.com, miaodecheng@sgu.edu.cn.

Y. Cheng is with the Department of Information Science and Engineering, Hunan University of Humanities, Science and Technology, Loudi Hunan 417000, China E-mail: yuncheng@huhst.edu.cn.

Jie Wu is with the Department of Computer and Information Sciences, Temple University, USA E-mail: jiewu@temple.edu.

A. Key pre-distribution

For the convenience of description, t-degree binary variable symmetric polynomials are abbreviated as key spaces. A key pool consists of *n*(*n* is a system parameter) key spaces. The i-th key space is constructed using the following formula:

$$K_i(x,y) = \sum_{j_1,j_2=0}^{t} a_{j_1,j_2} x^{j_1} y^{j_2} \tag{1}$$

where $a_{j_1,j_2} = a_{j_2,j_1}$, t is a system parameter.

The pre-distribution key $K_i(x,y)$ for node A can be calculated by using the following formula:

$$K_i(ID_A, y) = \sum_{j=0}^{t} b_j y^j \tag{2}$$

Node *A* selects *m* (*m* is a system parameter) key spaces from the key pool without repetition, and the IDs of these *m* key spaces can be calculated using the following formula:

$$KID_i = \left(ID_A \times m\right)\%n + i \quad (0 \le i < m, 0 \le ID_A < N) \tag{3}$$

Where *N* is the number of nodes in the network, which should meet the following conditions:

$$N \le \left\lfloor \frac{t \times n}{m} \right\rfloor \tag{4}$$

B. Shared keys establishment

The establishment of shared keys between nodes includes the following three steps, which will be described in detail by using two adjacent nodes A and B as examples:

Step 1. After the network is deployed, node A broadcasts its ID with radius *R* to discover its neighboring nodes. After receiving the broadcast information, neighboring node B can determine whether there is a common key space between them according to the formula (3). If there exists the key space $K_i\left(x,y\right)$, *B* only need to substitute A's ID into the formula (2) to obtain their shared key: $K_i(ID_A, ID_B) = K_i(ID_B, ID_A)$; Otherwise, if $ID_A < ID_B$, *B* will add *A* to its shared key establishment request set S_B; Otherwise, *A* will add *B* to its shared key establishment request set S_A.

Step 2. Node *A* broadcasts the following shared key establishment information with radius $w \times R$ (*w* represents the expansion factor): $Req_A = \left\{ID_A, S_A\right\}$. When its neighboring node *D* receives Req_A , if *D* shares common key spaces with both *A* and *C* ($C \in S_A$), then *D* becomes a participating node for *A*'s path key establishment. *D* first generates a random number rnd as the shared key between *A* and *C*, and then calculates the shared key establishment information according to the following formula:

$$\inf_D = \left\{ID_A, ID_D, E_{SK_{D-A}}\left(rnd\right), E_{SK_{D-C}}\left(rnd\right), H_{rnd}\left(rnd\right)\right\} \tag{5}$$

where *H*

where H_K() is a one-way hash function with the key *K*, E_K() is an encryption function with the key *K*. Finally, *D* sends $\inf_D$ to *A*. When *A* receives $\inf_D$, it calculates the shared key SK_{A-D} with *D* and decrypts it to obtain rnd' . If $H_{rnd'}\left(rnd'\right) \ne H_{rnd}\left(rnd\right)$, it discards rnd; otherwise, *A* stores rnd and sends the following message to *C*:

$$\inf_{A-C} = \left\{ID_C, ID_D, ID_A, E_{SK_{D-C}}\left(rnd\right), H_{rnd+1}\left(rnd+1\right)\right\} \tag{6}$$

When *C* receives $\inf_{A-C}$, it calculates the key SK_{C-D} shared with *D* and decrypts $\inf_{A-C}$ obtaining rnd' . If $H_{rnd'+1}\left(rnd'+1\right) \ne H_{rnd+1}\left(rnd+1\right)$, it discards rnd; otherwise, rnd becomes the shared key between *A* and *C*. If there are multiple shared keys between *A* and *C*, rnd_1, ..., rnd_q , then the shared key between them is $SK_{A-C} = rnd_1 \oplus \cdots \oplus rnd_q$, where $\oplus$ is the XOR operation.

Step 3. After the above 2 steps, if *C, the neighbor of A*, still cannot establish a shared key with *A*, it will be added to *A*'s second shared key establishment request set S'_A . Node *A* broadcasts the second shared key establishment information with radius *R*: $Req'_A = \left\{ID_A, S'_A\right\}$. When the common neighbor *E* of *A* and *C* ($C \in S'_A$) receives the above message, and if *E* shares keys with both *A* and *C* at the same time, *E* generates a random number rnd and securely sends it to *A* and *C*, respectively.

III. Scheme Analysis

The performance of random key pre-distribution schemes is usually measured by the following three parameter values [3]-[11]: local connectivity, resilience against node capture attacks and global connectivity. In our analysis and simulations, we use the following setups: we assume that the deployment area is flat; the neighbor relationship between SNs is symmetrical; that is, if *A* is a neighbor of *B*, then *B* is also a neighbor of *A*; the number of nodes in the network is 6000.

A. Computing local connectivity

The local connectivity, named *PL*, refers to the probability of two neighboring nodes establishing a shared pairwise key. In our scheme, the establishment of a shared key involves the above three steps. In the i-th step, PL_i represents the probability of establishing a shared key between two neighboring nodes, *PL* can be calculated using the following formula:

$$PL = PL_1 + \left(1 - PL_1\right)\cdot PL_2 + \left(1 - PL_1 - PL_2\right)\cdot PL_3 \tag{7}$$

where PL_1, PL_2 and PL_3 can be calculated using the formula (6), (7) and (8), respectively:

$$PL_1 = 1 - \frac{\binom{n-m}{m}}{\binom{n}{m}} \tag{8}$$

$$PL_2 = 1 - \left(1 - \left(PL_1\right)^2\right)^{Avg \times w^2} \tag{9}$$

where *Avg* represents the expected value of the number of neighbors of a node in the network.

In step 3, the probability that a node can participate in establishing a shared key is: $\left(PL_2\right)^2 + 2\cdot PL_1 \cdot PL_2$. Therefore, PL_3 can be estimated by using the following formula:

$$PL_3 = 1 - \left(1 - \left(PL_2\right)^2 - 2\cdot PL_1 \cdot PL_2\right)^{Avg'} \tag{10}$$

where Avg' represents the expected number of common neighboring nodes between two nodes, which can be calculated using the following formula：

$$Avg' = \frac{\pi R^2 + \frac{2}{3}\pi R^2 - \frac{3}{2}\sqrt{3}R^2}{2\pi R^2} \times Avg \approx 0.42Avg \qquad (11)$$

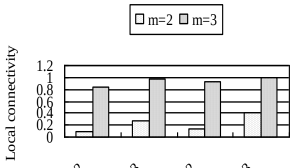


Fig. 1 Local connectivity as a function parameters Avg, m and w

The probability that two nodes can directly establish a shared key when *n* remains constant, namely PL_1, it increases with the increase of *m*. From formulas (7) to (11), it can be concluded that when *Avg* and *w* remain constant, *PL* increases with the increase of *m*. As shown in Fig. 1, when *W*=3, *Avg*=30 and *m* increases from 2 to 3, *PL* increases from approximately 0.42 to 0.99. When PL_1 remains constant, the number of nodes participating in path key establishment, namely N_1, the larger, the higher the probability that the two nodes can successfully establish a path key. As N_1 increases with the increase of *Avg* and *w*. Therefore, PL increases with the increase of *Avg* and *w*. As shown in Fig. 1, when *m*=3, *w*=3 and *Avg* increases from 25 to 30, *PL* increases from about 0.98 to 0.99. When *m* is small, increasing *w* can improve *PL* (see formula (9)). As shown in Fig. 1, when m=3, Avg=30 and *w* increases from 2 to 3, *PL* increases from about 0.92 to 0.99.

B. Computing resilience against node capture attack

Resilience against node capture attack, namely Pr, by estimating the fraction of total network communications which are compromised by a capture of *C* nodes not including the communications in which the compromised nodes are directly involved. Obviously, the smaller the Pr, the better the resilience. If Pr=0, then this scheme has perfect resilience. In the process of calculating Pr, like other schemes [1] - [11], it is assumed that hash functions and symmetric encryption are secure (secure and efficient hash functions and symmetric encryption in sensor networks have been studied by scholars [13]). Therefore, Pr can be calculated using the following formula：

$$\Pr = \frac{PL_1 \times P_{C_1} + PL_2 \times P_{C_2} + PL_3 \times P_{C_3}}{PL} \qquad (12)$$

In this scheme, each key space is allocated no more than t times, therefore, $P_{C_1}=0$. Increasing t will increase the computation overhead of nodes. In scheme [11], the value of *t* does not exceed 100. From the previous analysis, it can be seen that when *t* remains constant, increasing *N* can be achieved by increasing *n* and *w*, and reducing *m*. P_{C_2} and P_{C_3} can be calculated using the formula (12) and (14), respectively:

$$P_{C_2} = \left(\frac{C}{N}\right)^{E_2} \qquad (13)$$

where E_2 can be calculated using the following formula:

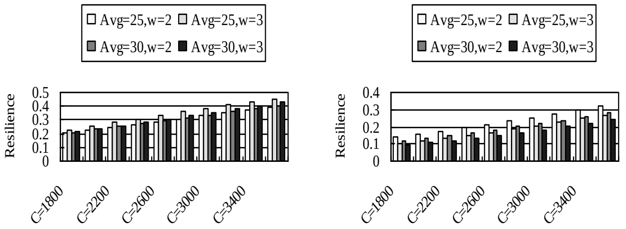
$$E_2 = \left(PL_1\right)^2 \times Avg \times w^2 \qquad (14)$$

$$P_{C_3} = 2\frac{PL_1 \times PL_2}{PL_3}\left(1-\left(\frac{C}{N}\right)^{E_2+E_3}\right) + \frac{\left(PL_2\right)^2}{PL_3}\left(1-\left(\frac{C}{N}\right)^{2E_2}\right) \qquad (15)$$

where E_3 can be calculated using the following formula:

$$E_3 = 0.42 \times PL_1 \times Avg \qquad (16)$$

From formulas (13) to (16), it can be concluded that the Pr will significantly decrease with the increase of PL_1, E_2 and E_3. And the above the values of three parameters will increase with the increase of m, Avg and w. Therefore, Pr will decrease with the increase of m, Avg and w. This can be confirmed from Fig.2.

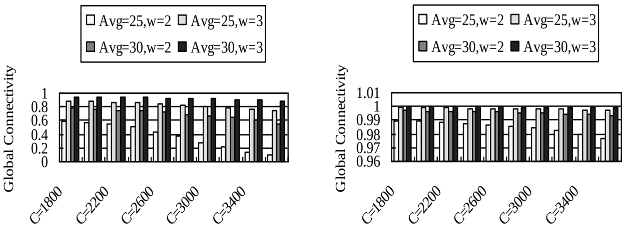


(a) m=2 (b) m=3

Fig. 2 Resilience as a function parameters Avg, m and w

C. Global connectivity

When the local connectivity is not 1, the network will be divided into multiple secure subsets. In each subset, there must be at least one secure path between any two nodes. If the maximum security subset only has 98% of the nodes in the network, then the remaining 2% of the nodes in the network are wasted due to the inability to communicate securely with the nodes in the maximum security subset. In this paper, we only simulated the global connectivity. From Fig. 3, it can be concluded that as long as the parameters are set properly, the global connectivity is high. For example, when m=3, w=3, Avg=30, and C=3600, the global connectivity is approximately 0.999.



(a) m=2 (b) m=3

Fig. 3 Global connectivity as a function parameters *Avg*, *m* and *w*.

D. Comparisons

In this subsection, we mainly compare the performance of our scheme with the PKPS scheme [11]. In our simulations, in the PKPS scheme, the size of the key pool, the number of nodes and the storage overhead of nodes are the same as our scheme. In PKPS, the establishment of shared keys includes two stages: shared key establishment and path key establishment. In the path key establishment stage, two adjacent nodes which cannot directly establish a shared key establish a key path through flooding. The two nodes on the key path must be adjacent and have a shared key. In these existing schemes, this method is used to establish shared keys [3] - [11]. These types of schemes have the following drawbacks: when the number of keys pre-distributed to nodes is small, the probability of establishing shared keys between nodes is very low. As shown in Table 1, when m=3, the *PL* of the PKPS scheme is only about 0.13. Although *PL* can be improved by increasing *m*, it can be observed from Fig. 4 that the resilience significantly decreases with the increase of *m*. As *m* increases from 3 to 5 and C=3000, the Pr of the PKPS scheme increases from about 0.3 to 0.999. As Pr increases, the PG of the PKPS scheme significantly decrease. As shown in Fig. 5, when C=3600, m=3, 4 and 5, PG is about 0.19, 0.02 and 0.004, respectively. In our scheme, when m=3 and w=3, PL≈0.995; when m=3, w=3 and C=3600, Pr≈0.24 and PG≈0.999.

Table 1 Comparison of PL between PKPS and our scheme

	PKPS(m=3)	PKPS(m=4)	PKPS(m=5)	mine
PL	0.12648	0.317522	0.578025	0.99498

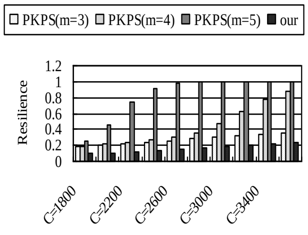


Fig. 4 Comparison of resilience between PKPS and our scheme

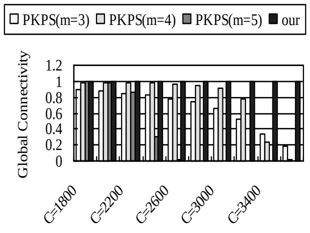


Fig 5. Comparison of Global connectivity between PKPS and our scheme

IV. Conclusion

In existing key pre-distribution schemes, when the pre-distribution key information of nodes is limited, the entire network is no longer securely connected. This paper proposes a new method for key pre-distribution and shared key establishment to address the problem. When there are fewer keys pre-distributed to a node before deployment, this method can achieve perfect resilience of shared keys established directly between nodes, high probability of establishing shared keys between nodes and secure connectivity for the majority of nodes in the network. For example, when n=180, t=100, m=3, w=3 and C=3600, PL≈0.995, Pr≈0.24 and PG≈0.999.

V. Acknowledgments

This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation, China, under Grant 2020A1515010923; in part by the Key Construction Discipline Scientiffc Research Ability Improvement Project of Department of Education of Guangdong Province, China, under Grant 2021ZDJS068; in part by the Talent Introduction Project of Shaoguan University, China, under Grant 99000618 and Grant 99000619.

References

[1] C. Karlof, D. Wagner, “Secure routing in wireless sensor networks: attacks and countermeasures,” in: IEEE SNPA’03, 2003.

[2] S. Zhu, S. Setia, S. Jajodia, “LEAP+: efficient security mechanisms for large-scale distributed sensor networks, ” ACM Transactions on Sensor Networks, vol. 2, no. 4, pp. 500-528, 2006.

[3] Eschenauer L, Gligor VD. “A key-management scheme for distributed sensor networks. Conference on Computer and Communications Security.” Proc. the 9th ACM Conference on Computer and Communications Security, pp. 41-47, 2002.

[4] H. Chan, A. Perrig, and D. Song, “Random Key Predistribution Schemes for Sensor Networks,” IEEE Symp. Security and Privacy, pp. 197-213, 2003.

[5] L. J. Kittur, A. R. Pais, “Combinatorial Design Based Key Pre-distribution Scheme with High Scalability and Minimal Storage for Wireless Sensor Networks,” Wireless Personal Communications, 128:855 – 873, 2023.

[6] R. a. Maciej, S. Alexander, “A key distribution technique for wireless sensor networks using spanning trees,” Expert Systems With Applications, 257, 124997, 2024.

[7] B. Zhou, S. Li, W. Wang, J. Wang, Y. Cheng, and J. Wu, “An effient authentication scheme based on deployment knowledge against mobile sink replication attack in UWSNs,” IEEE Internet Things Journal, vol. 6, no. 6, pp. 9738–9747, 2019.

[8] S. Li, B. Zhou, Q. Hu, et al., “A Secure Scheme Based on One-Way Associated Key Management Model in Wireless Sensor Networks,” IEEE Internet Things Journal, Vol. 8, No. 4, pp. 2920-2930, 2021.

[9] B. Zhou, S. Li, J. Wang, et al., “A secure model against mobile sink replication attacks in unattended sensor networks,” computer networks, 221: 109529, 2023.

[10] A. Msolli, N. Ajm, A. Helali, et al., “New key management scheme based on pool-hash for WSN and IoT,” Journal of Information Security and Applications, vol. 73:103415, 2023.

[11] W. Du, J. Deng, Y.S. Han, and P.K. Varshney, “A Pairwise Key Predistribution Scheme for Wireless Sensor Networks,” Proc. the 10th ACM Conf. Computer and Comm. Security (CCS), pp. 42-51, 2003.

[12] C. Blundo, A. D. Santis, A. Herzberg, “Perfectly-secure Key Distribution for Dynamic Conferences,” Information and Computation, 164(1): 1-23, 1998.

[13] O. A. Khashan, R. Ahmad, N. M. Khafajah, “An automated lightweight encryption scheme for secure and energy-efficient communication in wireless sensor networks,” Ad Hoc Networks, vol. 115, 102448, 2021.