A Secure Scheme based on One-way associated key management model in Wireless Sensor Networks

Sujun Li, Boqing Zhou, Qinqin Hu, Jianxin Wang, Senior Member, IEEE, JingGuo Dai, Weiping Wang, Member, IEEE, HuiYong Yuan, Yun Cheng, Jie Wu, Fellow, IEEE

***Abstract*—To achieve security in wireless sensor networks (WSNs), it is important to be able to encrypt messages sent among sensor nodes by using shared keys between them. Due to resource constraints, achieving such key agreement in wireless sensor networks is non-trivial. Previous research indicates that key management schemes using deployment knowledge can significantly improve the performance of WSNs. Nevertheless, in these schemes, resilient local connectivity and resilient global connectivity become unstable when deployment error changes. To resolve the above problem, in this paper, a one-way associated key management model is proposed. In this model, the key pool consists of two layers: the global layer and the local layer. According to different deployment errors, the number of keys allocated from the global key pool and local key pools can be dynamically adjusted, thereby improving the stability of networks' performance. In multi-phase sensor networks, analysis and simulation indicate that our scheme has better adaptability in applications where deployment error changes as compared with related schemes.**

***Index Terms*— wireless sensor networks, secure, one-way associated key management model, deployment error.**

This work is partially supported by the Research Foundation of Education Committee of Hunan Province, China (Grant No. 16A110), the Natural Science Foundation of Guangdong Province, China (Grant No. 2020A1515010923), the National Natural Science Foundation of China (Grant No. 61672543, 61772559, 61571188), the talent Introduction Project of Shaoguan University, China (Grant No. 99000618, 99000619). *(Corresponding authors: Jianxin Wang; Boqing Zhou.)*

S. Li and B. Zhou are with the School of Information Engineering, Shaoguan University, Shaoguan 512005, China, and also with MOE-LCSM, School of Mathematics and Statistics, Hunan Normal University, Changsha, Hunan 410081, China (e-mail: lsj_paper@163.com, zbq_paper@163.com).

Qinqin Hu is with the School of Information Science and Engineering, Hunan Institute of Science and Technology(e-mail: 1924667824@qq.com)

W. Wang and J. Wang are with the School of computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: wpwang@mail.csu.edu.cn; jxwang@mail.csu.edu.cn).

JingGuo Dai and Huiyong Yuan are with the School of Information Engineering, Shaoguan University, Shaoguan 512005, China (e-mail: 646257139@qq.com, 303882171@qq.com).

Y. Cheng is with the School of Information, Hunan University of Humanities, Science and Technology, Loudi 417000, China (e-mail: chy8370002@gmail.com).

J. Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122 USA (e-mail: jiewu@temple.edu).

I. INTRODUCTION

WSNs usually consist of a large number of ultra-small autonomous devices. Each device, called a sensor node, is battery powered and equipped with integrated sensors, data processing, and short-range radio communication capabilities. WSNs are being deployed for a wide variety of applications. When WSNs are deployed in a hostile environment, security becomes extremely important as they are vulnerable to different types of malicious attacks [1]. Hence, it is important to protect communications among sensor nodes to maintain message confidentiality and integrity. As one of the most fundamental security services, pairwise key establishment enables sensor nodes to communicate securely with each other by using cryptographic techniques.

Public-key operations consume energy approximately three orders of magnitude higher than symmetric key encryption [2]. Therefore, in the last few years, different key pre-distribution schemes using symmetric key algorithms have been developed for WSNs [3-22].

A. Motivation

To improve the performance of WSNs Du et al. proposed the first key management scheme based on deployment knowledge [14]. In [14], a target field is partitioned into square grids. Yu et al. proposed a new key management scheme [15]. In [15], a target field is partitioned into hexagons. Fanian et al. re-divided hexagonal cells and proposed four deterministic key management models [16]. When deployment error is small, the performance of [15] can be improved. However, none of these schemes takes into account multi-phase WSNS. In WSNS, in order to maintain the normal operation of networks throughout their life cycles, new nodes must be added to them multiple times to replace sensor nodes which have died or have been captured. In [14-16], new added nodes select keys from the same key pool with nodes have been deployed, the capture of a node will increase fraction of keys known to the adversary. When a certain number of nodes are captured, the adversary has enough keys to compromise a large number of links making the network ineffective. Therefore, adding new nodes to the network with keys from the same key pool will not help because the keys stored in these nodes already have been compromised. Zhou et al. proposed two key management schemes based on deployment knowledge for multi-phase WSNS [17-18].

Schemes in [14-18], networks' performance decreases significantly as deployment error increases. In [19], Zhou et al. proposed a layered key management scheme, namely SS-LM. This scheme can adapt to different deployment errors by setting different key association layers *T*. When *T*=1, the above key management model degenerates into the YG scheme [16]. When deployment error is large, by increasing the parameter *T* properly, the performance of these schemes in [14-18] can be improved significantly. However, this model has the following drawback: once the parameter *T* is determined in the 1st phase, it is difficult to change it in subsequent phases. In other words, it is difficult to improve networks' performances by setting appropriate values of parameters according to the specific situation of each deployment phase. Therefore, for multi-phase WSNS, key management schemes based on deployment knowledge, which can provide good and stable performance when deployment errors change widely and dynamically, still needs to be studied.

B. Main contribution of our scheme

In this paper, to improve the scheme's adaptability to dynamic changes of deployment errors, we construct a new one-way associated key management model. The main contributions of our work are the followings:

1. A one-way associated key management model is proposed. In this model, the key pool consists of two layers: the global layer and the local layer. Keys in local key pools can be calculated using keys in the global key pool, otherwise it is not.

2. As the deployment error increases, the probability that two nodes whose deployment points are neighboring become actual neighbors will decrease. When the pre-distribution key information remains unchanged, the probability of establishing a shared key between the nodes will decrease. To solve this problem, a scheme, where the number of distribution keys from the global key pool and local key pools can be dynamically adjusted, is proposed. Analysis and simulation show that the proposed scheme has good adaptability to the dynamic changes of deployment error in multi-phase WSNs.

C. Organization

The rest of this paper is organized as follows. At first, the background of our scheme is presented in Section II. Subsequently, the proposed scheme will be presented in Section III. Together with a comprehensive comparison with a known scheme, the theoretical and experimental results will be described in Section IV. At last, the conclusion will be made in Section V.

II. Related work

To improve the performance of pairwise key establishment, Du et al. [14], Yu and Guan (YG scheme) [15] developed a scheme using pre-deployment knowledge, respectively. In [14], the network area is divided into a grid, sensors are deployed in groups, each group has a single deployment point which locates in a square cell, and the pdfs (probability distribution functions) of the final resident points of all sensors in a group are the same. A global key pool is partitioned into many local key pools and each square grid corresponds to a local key pool. Nodes deployed in a square grid pick keys from the local key pool of the grid. The scheme can substantially improve networks' connectivity, resilience against node capture, and lower the storage overhead as compared with these schemes not using deployment knowledge. In [15], the network area is divided into hexagonal cells. Compared with [14], the scheme achieves a higher connectivity with fewer distribution keys and a shorter transmission range. Fanian et al. re-divided hexagonal cells and proposed four key management models [16]. When the deployment error is small, the scheme can improve the performance of the scheme in [15]. In these schemes [14-16], networks' security throughout their lifecycle is not taken into account. Zhou et al proposed a key management scheme, namely ESPK, based on the combination of one-dimension key chains and deployment knowledge [17]. The scheme can provide high network security throughout their lifecycle. However, local connectivity of the ESPK decreases with the increase of the deployment phase. The above deficiency is improved by the scheme in [18].

For applications with large deployment error, Zhou et al. proposed the SS-LM scheme [19]. In this scheme, each cell is a basic cell, each basic cell has *T*-layer association cells which are around it, and shared keys between a basic cell and its *T*-layer association cells are established by using three-dimension backward key chains. In the model, the larger *T* is, the more association cells of a basic cell are. When *T*=1, the model degenerates into the model of YG scheme [15]. By increasing *T*, this solution can be applied to applications with large deployment error. However, this solution has the following drawback: once the value of *T* is determined in the 1st phase, it is difficult to change it in the subsequent phases. For example, in the 1st phase and the 2nd phase, *T* should be equal to 1 and 2, respectively, which will cause nodes deployed in the 2nd phase to be unable to establish shared keys with nodes deployed in the 1st phase using keys from the 2nd layer contact cells. This results in an increase in the number of isolated nodes which cannot establish shared keys with their neighbor nodes and a decrease in the resilient global connectivity.

III. Our Scheme

Our scheme includes the following four parts: deployment model, one-way associated key management model, key pre-distribution method, and shared key establishment.

A. Deployment model

In our scheme, the deployment model is the same as [19]. As shown in Fig. 1, a target field is partitioned into hexagon cells, and each cell has a deployment point that resides in the center of the cell. Network deployment includes many phases, $G_{(r,c)}$ represents the set of groups whose deployment points locate in the cell (r,c) (For the convenience of description, in the following chapters, nodes within a cell refer to nodes whose deployment points are located in this cell rather than nodes which are actually located in this cell after deployment, except for special instructions), and $G^{i}_{(r,c)}$ represents the i^{th} phase subgroup of the group $G_{(r,c)}$. Node distribution follows two-dimensional Gaussian distributions

with the deployment point as center, as follows:

$$f_{(r,c)}^{i}(x,y)=\frac{1}{2\pi\left(\sigma^{i}\right)^{2}}e^{-\left[(x-x_{(r,c)})^{2}+(y-y_{(r,c)})^{2}\right]/2\left(\sigma^{i}\right)^{2}} \quad (1)$$

where σ^i is the standard deviation of distribution. We know that the distance between a resident point and the deployment point is less than $3\sigma^i$ with probability 0.9987. When the length of a hexagon cell is fixed, the probability that nodes, whose deployment points locate in neighboring cells, become neighbors will gradually decrease with the increases of σ^i. In WSNS, σ^i is affected by various factors such as the height of a helicopter and the weather when nodes are deployed, etc. For example, when nodes are deployed in hostile environments, it is necessary to increase the height of helicopter and protect nodes through the help of parachutes. It's the same as SS-LM [26], σ^i represents deployment error. The larger the σ^i, the greater the deployment error is.

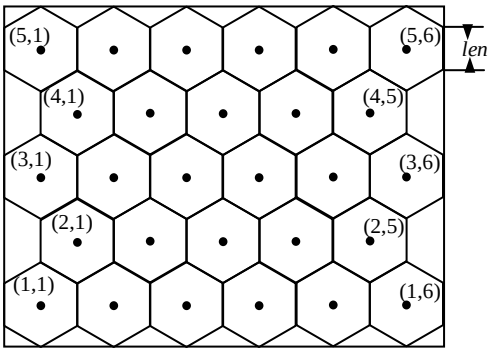


Fig. 1. A target field is partitioned into hexagon grids. • represents a deployment point.

B. *One-way associated key management model*

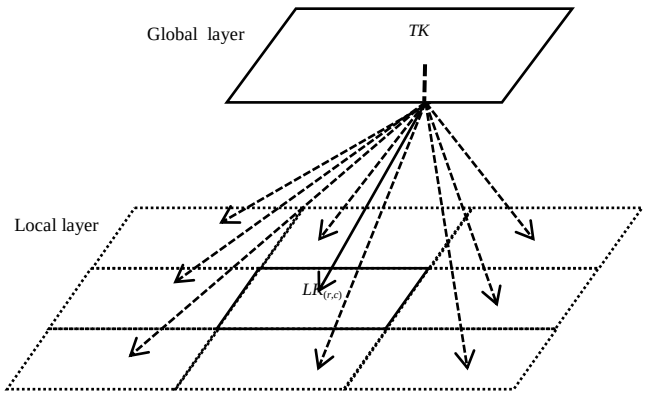


Fig. 2. Hierarchical Key Management Model

As shown in Fig. 2, the one-way associated key management model consists of global and local layers, denoted by *PG* and *PL*, respectively. The local key pool of the cell (*r*,*c*) is represented by $PL_{(r,c)}$. And there are some shared keys in two adjacent local layers. Each key of the global layer is used for calculating a sub-key for each local key pool. As shown in Fig. 2, the sub-key $LK_{(r,c)}$ of the local key pool $PL_{(r,c)}$ is calculated by using the key *TK* in the global key. Therefore, all keys in the local key pool can be calculated quickly by using keys in the global key pool, but it is computationally infeasible to calculate keys in the global key pool by using keys in the local key pool.

In this paper, the two-layer key management model is implemented using three-dimension backward key chains [19]. Of course, it can also be implemented using other technologies. For the fairness of subsequent comparisons, we adopt this approach. In this implementation mode, the key pool is divided into generations according to the deployment phase, and PG^i and $PL_{(r,c)}^i$ represent *PG* and $PL_{(r,c)}$ of the i^{th} phase, respectively.

The construction method of a three-dimension backward key chain is same as [19]. The detailed description is as follows:

(1) The 1^{st} dimension keys, namely k_j^i, is generated by a one way hash function, namely $H_1()$, with a generation key g_j, as follows:

$$k_j^i=H_1(k_j^{i+1})\ (k_j^n=H_1(g_j),\ 1\le i\le n-1) \quad (2)$$

(2) The 2^{nd} dimension key chain, namely $k_j^{(i,l_2)}$, is generated by a one way keyed hash function, namely H(), with the key k_j^i and a random seed l_2 as follows:

$$k_j^{(i,l_2)}=H(k_j^i,l_2) \quad (3)$$

(3) The 3^{rd} dimension key chain, namely $k_j^{(i,l_2,l_3)}$, is generated by a $H()$, with the key $k_j^{(i,l_2)}$ as follows:

$$k_j^{(i,l_2,l_3)}=H(k_j^{(i,l_2)},k_j^{(i,l_2,l_3-1)})\ (k_j^{(i,l_2,1)}=H(k_j^{(i,l_2)},0),\ 1\le l_3\le L_3) \quad (4)$$

The key set of PG^i is: $\left\{k_j^i \middle| 1\le j\le m\right\}$, $PL_{(r,c)}^i$ consists of the following two parts: one is local ordinary key pool, namely $PLC_{(r,c)}^i$. Its key set is:

$\left\{k_j^{(i,l_2,l_3)} \middle| 1\le j\le m, l_2=r\,\|\,c, 1\le l_3\le L_3\right\}$, where $\|$ indicates information connection operation. The other is local generation key pool, namely $PLG_{(r,c)}^i$. The composition of its key set is as follows:

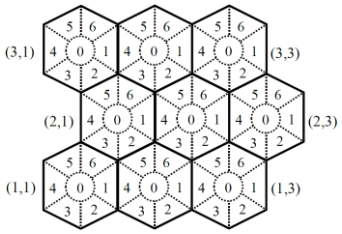


Fig. 3. Local key pools of the cell (2, 2)

For a cell (r, c), keys in $LG_{(r,c)}^i=\left\{k_j^{(l_2,i)} \middle| 1\le j\le m, l_2=r\,\|\,c\right\}$ are separated into 7 equal parts, a part is denoted by $(LG_{(r,c)}^i)_x$ $(0\le x\le 6)$. All $(LG_{(2,2)}^i)_x$ are ordered according to the method presented in Fig. 3. Then these neighbor cells exchange their key sets with each other. For example, the cell (2, 2) gives $(LG_{(2,2)}^i)_1$ to the cell (2,3), the cell (2,3) gives $(LG_{(2,3)}^i)_4$ to the cell (2,2). So we can have

$$PLG_{(2,2)}^i=(LG_{(2,2)}^i)_0\cup(LG_{(2,3)}^i)_4\cup(LG_{(1,3)}^i)_5\cup(LG_{(1,2)}^i)_6\cup(LG_{(2,1)}^i)_1\cup(LG_{(3,2)}^i)_2\cup(LG_{(3,3)}^i)_3.$$

C. Key pre-distribution

A sensor node $a^i_{(r,c)}$ ($a^i_{(r,c)} \in G^i_{(r,c)}$) is preloaded with $t1$, t2 and $t3$ ($t3>>t1+t2$) keys along with these keys' IDs, from PG^i, $PLG^i_{(r,c)}$ and $PLC^i_{(r,c)}$, respectively, which meets the following condition: the number of keys from a three-dimension backward hash key chain is no more than 1. For example, if the key k^i_j of the key chain j has been pre-distributed to $a^i_{(r,c)}$, then the 2nd dimension and the 3rd dimension keys of the key chain cannot be pre-distributed to $a^i_{(r,c)}$.

D. Shared key establishment

In our scheme, after shared key establishment, pre-distribution keys of nodes are hashed. That is, if $a^i_{(r,c)}$ is pre-distributed the key $k^{(i,l_2)}_j$, after shared key establishment, the node stores the following key: $Hk^{(i,l_2)}_j = H(k^{(i,l_2)}_j, ID_{a^i_{(r,c)}})$, where $ID_{a^i_{(r,c)}}$ is the identity of the node $a^i_{(r,c)}$.

Next, the method for two nodes $a^{i_1}_{(r_1,c_1)}$ ($a^{i_1}_{(r_1,c_1)} \in G^{i_1}_{(r_1,c_1)}$) and $b^{i_2}_{(r_2,c_2)}$ ($b^{i_2}_{(r_2,c_2)} \in G^{i_2}_{(r_2,c_2)}$) establishing a shared key is described in detail.

Their common keys can be calculated according to the following cases:

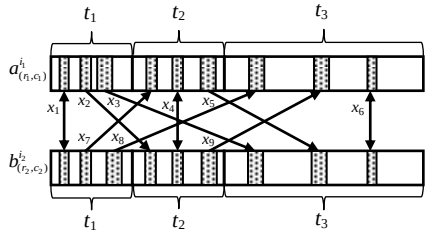


Fig. 4. Shared keys between two nodes when $(r_1,c_1)=(r_2,c_2)$ and $i_1=i_2$.

1. when $i_1=i_2$, as shown in Fig. 4, common keys of $a^{i_1}_{(r_1,c_1)}$ and $b^{i_2}_{(r_2,c_2)}$ consist of the following three parts: 1. x_1 keys from PG^{i_2}; 2. x_2 and x_7 common keys from $PLG^{i_2}_{(r_2,c_2)}$ and $PLG^{i_1}_{(r_1,c_1)}$, respectively; 3. x_3 and x_8 common keys from $PLC^{i_2}_{(r_2,c_2)}$ and $PLC^{i_1}_{(r_1,c_1)}$, respectively. **If $(r_1,c_1)=(r_2,c_2)$,** the shared keys between them also includes the following three parts: 1. x_4 common keys from $PLG^{i_2}_{(r_2,c_2)}$; 2. x_5 and x_6 common keys from $PLC^{i_2}_{(r_2,c_2)}$; 3. x_9 common keys from $PLC^{i_1}_{(r_1,c_1)}$. **Otherwise, if $(r1,c1)$ and $(r2,c2)$ are neighbor cells**, the shared keys between them also include the following two parts: 1. x_5 common keys from $PLC^{i_2}_{(r_2,c_2)}$; 2. x_9 common keys from $PLC^{i_1}_{(r_1,c_1)}$. **Otherwise,** $x_4=x_5=x_6=x_9=0$.

2. $i_1>i_2$, $x_4=x_5=x_6=x_7=x_8=x_9=0$, the value of x_1, x_2, x_3 is same as the case 1.

If $x_1+x_2+x_3+x_4+x_5+x_6+x_7+x_8+x_9>0$, XOR all their common keys to get the key K_{ab}, and the shared key, namely PK_{ab}, between $a^{i_1}_{(r_1,c_1)}$ and $b^{i_2}_{(r_2,c_2)}$ is the key $PK_{ab} = H(K_{ab}, ID_{a^{i_1}_{(r1,c1)}} \oplus ID_{b^{i_2}_{(r_2,c_2)}})$.

IV. Performance and Security Evaluation

In this section, we analyze the performance and security of our scheme, including resilient local connectivity [20] and resilient global connectivity [21]-[22].

In our analysis and simulations, we use the following setups:

1. The area is divided into hexagon cells and the length of each hexagon cell is 50, namely *len*=50. The center of each cell is the deployment point (see Fig. 1). We assume that node deployment follows a two-dimension Gaussian distribution.

2. The wireless communication range for a node is 40m (R=40m).

3. The size of the global key pool, namely m, is 3500, and the length of the 3rd dimension of a three-dimension backward key chain based on deployment knowledge is 50 (L_3=50).

4. The presented experimental data is an average of 50 replicates.

A. Multi-phase attack model

In order to analyze the performance of the scheme, we need to construct an attack model. The attack model in this paper is similar to that in [14]. In [14], an adversary captures nodes randomly within a region. And the region is assumed to be a circle centered at point with coordinate (x,y) with radius R_c. However, in [14], the attack model is proposed for one-phase WSNs, and does not consider the situation where an adversary attacks the WSN multiple phases. Our solution can be applied to multi-phase WSNs, so a multi-phase attack model for the adversaries should be constructed.

In the multi-phase attack model, we suppose that captures occurring between the i^{th} phase and the $(i+1)^{th}$ phase are called the i^{th} time capture. If an attacker captures a sensor node, all key information it holds will also be compromised. And it is supposed that only a limited number of nodes may be compromised by an attacker during the short time period of the shared key establishment [3]-[12]. In order to evaluate the effectiveness of this scheme against capture attacks, we assume that: in the 1st phase, if an attacker captures nodes in a certain area, and in subsequent phases, he will continue to capture nodes in the area. The purpose of this assumption is to allow the attacker to obtain more keys from the corresponding local key pools. And it will pose a greater threat to the performance and security of WSNs.

In the following simulations, we assume that the value of R_c is 250, and the attacker captures 50% of the nodes in a certain area. When the number of nodes in a cell is less than 15, you need to add 30 new nodes to this cell.

B. Resilient local connectivity

Local connectivity is the probability that two neighboring sensor nodes can establish a shared key. Resilient local connectivity is the probability that two neighboring nodes can establish a secure shared key under the presence of attacks.

Resilient local connectivity in the I^{th} phase, namely RLC^I, can be estimated by the following formula:

$$RLC^I = PC^I \times \left(1\text{-}R^I\right) \tag{5}$$

Where PC^I and R^I indicate the probability that two neighboring nodes can establish a shared key after the I^{th} phase, and the probability that the shared key established between two un-captured neighbor nodes is compromised after the I^{th} capture, respectively.

1. Computing PC^I

The shared keys between nodes can be divided into the following two categories: one is established by using keys from global key pools, and the other is established by using keys from local key pools. There P_1^I and P_2^I represent the probabilities of establishing shared keys by using the above two categories of keys, respectively. PC^I can be calculated using the following formula:

$$PC^I = \left(1-\left(1-P_1^I\right)\times\left(1-P_2^I\right)\right) \tag{6}$$

The method for calculating the shared key shows that the value of P_1^I is independent of the deployment knowledge, and can be calculated using the following formula:

$$P_1^I = Pg_1 \times P_{SP} + Pg_2 \times \left(1-P_{SP}\right), \tag{7}$$

where Pg_1 and Pg_2 represent the probabilities of shared keys being established between two nodes which are deployed in the same phase and which are deployed in different phases, respectively. P_{SP} represents the proportion of nodes deployed in the same phase, whose value is related to the capture model and the addition model of new nodes, and is independent of the key management scheme. Therefore, we'll only analyze Pg_1 and Pg_2 in detail.

Pg_1 and Pg_2 can be calculated using formula (8) and formula (9), respectively:

$$Pg_1 = \sum_{q=1}^{2t_1} Pg_1^q \ , \tag{8}$$

where $Pg_1^q = \dfrac{\sum_{x_c=q} \binom{m}{x_c} \times \binom{m\text{-}x_c}{2(t_1+t_2+t_3-x_c)} \times \binom{2(t_1+t_2+t_3-x_c)}{t_1+t_2+t_3-x_c}}{\binom{m}{t_1}^2 \times \binom{m-t_1}{t_2+t_3}^2}$

$(x_c = x_1 + x_2 + x_3 + x_7 + x_8)$.

$$Pg_2 = \sum_{q=1}^{t_1} Pg_2^q \ . \tag{9}$$

where $Pg_2^q = \dfrac{\sum_{x_1+x_2+x_3=q} \binom{t_1}{x_1} \times \binom{t_2}{x_2} \times \binom{t_3}{x_3} \times \binom{m\text{-}t_1\text{-}t_2\text{-}t_3}{t_1-x_1-x_2-x_3} \times \binom{m-t_1}{t_2+t_3}}{\binom{m}{t_1} \times \binom{m-t_1}{t_2+t_3}}$.

From the key establishment process, we can see that the value of P_2^I is related to the deployment phase of two neighbor nodes and the cells where them are located. Therefore, the calculation of P_2^I is divided into the following 3 cases:

1. If two neighbor nodes are deployed in different phases, $P_2^I = 0$;

2. If the deployment cells of two neighbor nodes are not the same or are not adjacent, $P_2^I = 0$;

3. In addition to the above two cases, P_2^I can be calculated using the following formula:

$$P_2^I = P_{GS} \times P_{SP-GS} \times Pc_1 + P_{GN} \times P_{SP-GN} \times Pc_2 \ , \tag{10}$$

where P_{GS} and P_{GN} represent the probability of two nodes, whose deployment cells are the same and are adjacent, becoming neighbors after deployment, respectively. The values of the two parameters are related to the deployment error; P_{SP-GS} and P_{SP-GN} represent the proportion of these nodes, which are deployed in the same phase, and whose deployment cells are the same and are adjacent, respectively; Pc_1 and Pc_2 represent the probabilities of a shared key being established by using keys from local key pools only, between two nodes which are deployed in the same phase and whose deployment cells are the same and are adjacent, respectively.

Pc_1 and Pc_2 can be calculated using the following formulas (11) and (12), respectively:

$$Pc_1 = \sum_{q=1}^{2t_2} Pc_1^q \ , \tag{11}$$

where $Pc_1^q = \dfrac{\sum_{x_C=q} \sum_{x_{10}=0}^{t_3-x_5-x_6-x_9} \binom{t_2+t_3}{x_C+x_{10}} \times \binom{m\text{-}2t_1\text{-}t_2\text{-}t_3}{t_2+t_3-x_C} \times \binom{t_2+t_3-x_C}{t_2-x_4-x_5} \times \left(\dfrac{L_3-1}{L_3}\right)^{x_{10}} \times \binom{L_3}{1}^{-x_6}}{\binom{m-2t_1}{t_2+t_3} \times \binom{t_2+t_3}{t_2}}$

$(x_C = x_4 + x_5 + x_6 + x_9)$.

$$Pc_2 = \sum_{q=1}^{2t_2} Pc_2^q \ , \tag{12}$$

Where $Pc_2^q = \dfrac{\sum_{x_C=q} \binom{t_2+t_3}{x_C} \times \binom{m\text{-}2t_1\text{-}t_2\text{-}t_3}{t_2+t_3-x_C} \times \binom{t_2+t_3-x_C}{t_2-x_5}}{\binom{m-2t_1}{t_2+t_3} \times \binom{t_2+t_3}{t_2}}$ $(x_C = x_5 + x_9)$.

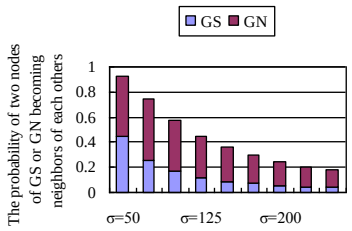
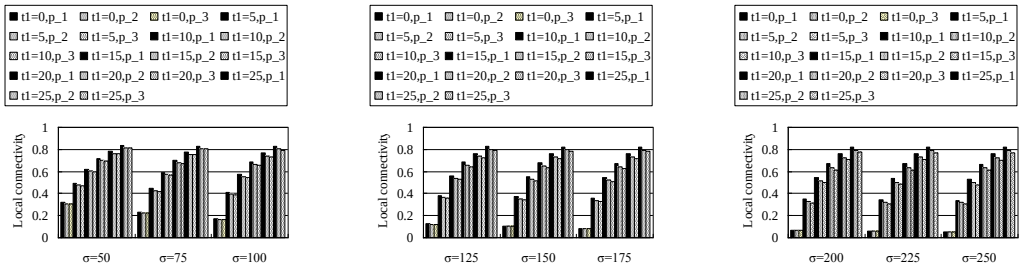


Fig. 5. P_{GS} and P_{GN} as a function parameter σ

Fig. 5 shows that the probability that two nodes, come from the same deployment cell or two neighbor deployment cells, become neighbor decreases significantly as the deployment error σ increases. For example, when the values of σ are 50 and 250, the values of P_{GS} + P_{GN} are about 0.93 and 0.17, respectively. That is, when the size of the deployment cells is fixed, the performance of networks cannot be improved only by using local key pools. If the size of deployment cells is expanded, the scheme based on deployment knowledge will gradually degenerate into the E-G scheme, and when the deployment error is small, it is also not feasible to improve



(a) (b) (c)

Fig. 6. Connectivity as a function parameters σ, t1,t2,t3. In these figures, p_i (0<i<4) represents the i[th] phase, t_1+t_2=30 and t_3=100.

network performance by using deployment knowledge.

When *t*1=0, this scheme will degenerate into schemes in [21]-[22], [24]-[25]. Fig.5 shows that when σ is 50 and 250, local connectivity in the first phase is about 0.32 and 0.05, respectively. It is clear that this situation does not meet the needs of applications with large variations in the deployment error. Hence, we need to dynamically adjust the value of *t*1 for applications with different deployment error.

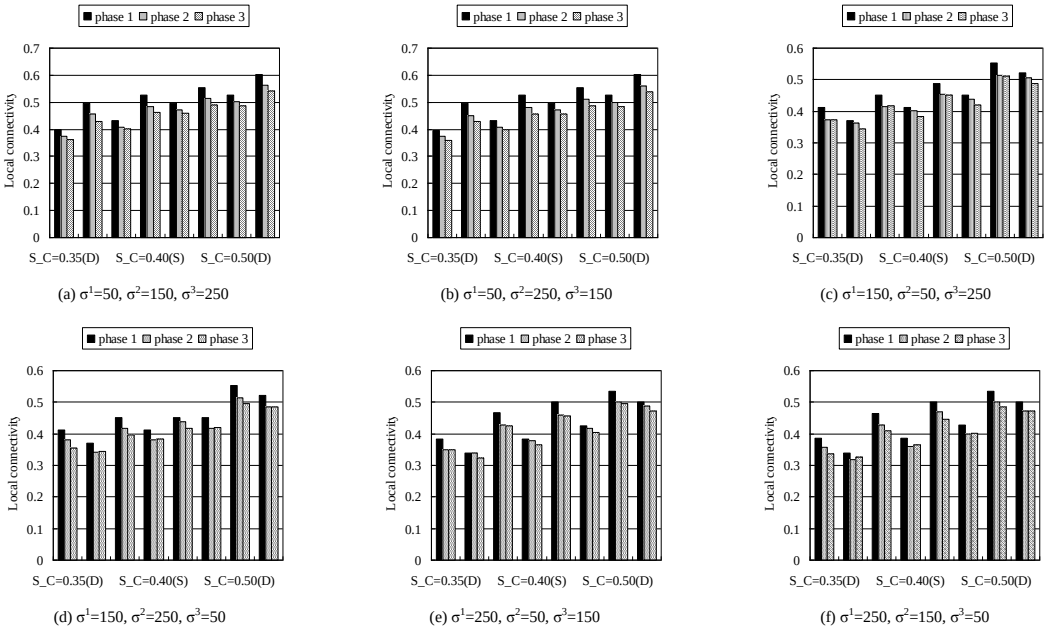
From formulas (6) to (13), the following conclusions can be drawn:

1. The probability of a shared key being established based on the parameter *t*1 is independent of deployment knowledge (see formula (7)). The greater its value, the greater the local connectivity and the lesser affected by the deployment error. (see formulas (8) and (9)). As shown in Fig. 6, when t1=5, t1=15 and t1=25, σ increases from 50 to 250, the local connectivity in the first phase is reduced by about 0.15, 0.05, and 0.01, respectively;

2. The probability of a shared key being established based on the parameters t2 and t3 is susceptible to the deployment error (see formula (10)). It is not difficult to find that the influence of parameter *t*1 on connectivity is the greatest, and the influence of parameter *t*3 on connectivity is the least (see formulas (11), (12) and (13)). As shown in Fig. 6, when σ=175, t1+t2=30, and when t2 decreases from 25 to 5, the local connectivity in the 1[st] phase increases from about 0.36 to about 0.82;

3. As deployment phase increases, the local connectivity decreases. However, as deployment phase increases, the ratio of nodes deployed in the same phase among the neighbor nodes tends to be stable, and the local connectivity will also become stable (see formula (10)). As shown in Fig. 6, when σ=175, t1=15 and t2=15, from the 1[st] phase to the 2[nd] phase, the local connectivity drops by about 0.03, however, from the 2[nd] phase to the 3[rd] phase, the connectivity decreases by about 0.01.

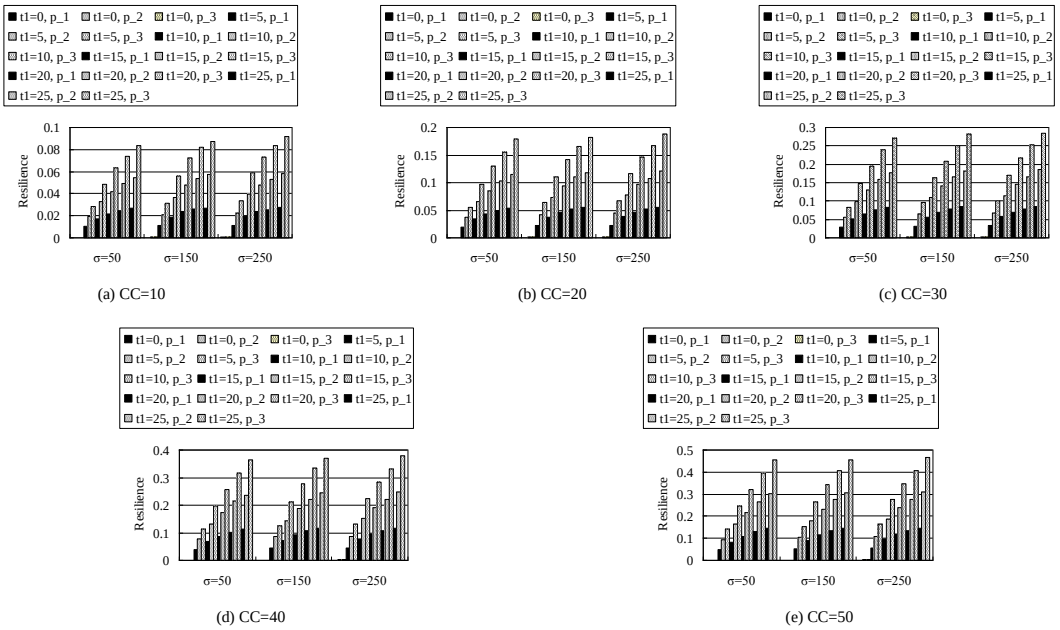
4. The influence of deployment error on network performance can be reduced by dynamically adjusting parameters t1, t2 and t3 according to the deployment error of each phase. Fig. 7 shows the effects of setting the values of parameters statically and dynamically on the connectivity, respectively. In Fig. 7, we consider all combinations of the values of deployment error are 50, 150, and 250 in the 1[st] phase, 2[nd] phase, and 3[rd] phase. In this simulation, t1+t2=30 and t3=100. In the dynamic adjustment method of parameters, the value of *t1* is dynamically adjusted according to the predetermined connectivity. In the static adjustment method of parameters, the value of *t1* is determined by the predetermined



(a) σ^1=50, σ^2=150, σ^3=250 (b) σ^1=50, σ^2=250, σ^3=150 (c) σ^1=150, σ^2=50, σ^3=250

(d) σ^1=150, σ^2=250, σ^3=50 (e) σ^1=250, σ^2=50, σ^3=150 (f) σ^1=250, σ^2=150, σ^3=50

Fig. 7. Comparisons of connectivity in parameters dynamic and static settings. In these figures, S represents that t1, t2 and t3 remain unchanged in all phases, and D represents that t1, t2 and t3 are set according to the actual situation in each phase. S_C represents the predetermined connectivity, and the other parameter settings are the same as in Fig. 6.



(a) CC=10 (b) CC=20 (c) CC=30

(d) CC=40 (e) CC=50

Fig. 8. Resilience as a function parameters CC, σ, t_1, t_2 and t_3. Parameter settings are the same as in Fig. 6.

local connectivity when σ^1=150. From Fig. 7, it can be concluded that the dynamic adjustment of parameters can make the network performance more stable than the static setting method of parameters. As shown in Fig. 7, in the following two cases: when σ^1=50, σ^2=150, σ^3=250 and σ^1=250, σ^2=150, σ^3=50, PC^3 (S_C=0.35) drops from about 0.43 to about 0.32, while in the dynamic adjustment method of parameters, PC^3 drops from about 0.36 to about 0.34.

2. Computing R^I

The construction method of three-dimension backward key chains shows that: 1. Keys of the key chain *j* can be calculated by using k_j^I if their deployment phases are not greater than *I* (see formulas (2) to (4)); 2. The third-dimension keys of the I^{th} phase of the key chain *j* can be calculated by using $k_j^{(I,I_2)}$ (see formulas (3) and (4)). Therefore, nodes, which are deployed in the I^{th} phase, are compromised, which will pose a threat to these shared keys established between nodes deployed in the I^{th} phase or deployed before the I^{th} phase.

In this paper, after shared keys establishment, all pre-distribution keys are hashed. Because shared keys establishment time is short, the assumption that only a few nodes are captured during this phase is reasonable [3]-[4], [12]-[19]. Here, CC^i represents the number of nodes, which are deployed in the i^{th} phase, are compromised before completing shared keys establishment. After *I* captures, the number of nodes which can calculate keys of key pools of the i^{th} phase, namely $CC^{i,I}$, can be calculated as follows:

$$CC^{i,I} = \sum_{i_1=i}^{I} CC^{i_1} \tag{13}$$

Hence, after *I* captures, the probability that keys of the global key pool PG^i are compromised should be:

$$PGR^{i,I} = 1-\left(1-\frac{t1}{m}\right)^{CC^{i,I}} \tag{14}$$

Supposing that a shared key between nodes is established by using *q* keys from the key pool PG^i. The probability of the shared key being compromised is:

$$PGR_q^{i,I} = \left(PGR^{i,I}\right)^q \tag{15}$$

The probability that keys of $PLG_{(r,c)}^i$ and $PLC_{(r,c)}^i$ are compromised can be calculated using formulas (16) and (17), respectively:

$$PLGR_{(r,c)}^{i,I} = 1-\left(1-\frac{t_1}{m}\right)^{CC^{i,I}} \times \left(1-\frac{t_2}{m}\right)^{CC_{(r,c)}^i} \tag{16}$$

$$PLCR_{(r,c)}^{i,I} = 1-\left(1-\frac{t_1}{m}\right)^{CC^{i,I}} \times \left(1-\frac{t_2}{m}\right)^{CC_{(r,c)}^i} \times \left(1-\frac{t_3}{m\times L_3}\right)^{CC_{(r,c)}^i} \tag{17}$$

Similar to the formula (15), we can get $PLGR_{(r,c)_q}^{i,I}$ and $PLCR_{(r,c)_q}^{i,I}$.

In summary, the value of R^I is related to the number of un-captured nodes, the number of nodes captured before the shared key establishment and the composition of shared keys (including the parameter *q*, and the number of keys from PG^i, $PLG_{(r,c)}^i$ and $PLC_{(r,c)}^i$, respectively). Here, R^I is simulated. In this simulation, we assume that CC^i in each phase is equal, which is represented by *CC* in the following descriptions.

From formulas (13) to (17), the following conclusions can be drawn:

1. When the values of parameters t1, t2, t3, and *CC* remain unchanged, R^I increases as σ increases. This is because as the σ increases, the probability of two nodes from the same cell or adjacent cells becoming neighbors decreases significantly (see Fig. 5). As a result, the proportion of shared keys established by using keys from local key pools decreases. That is, q decreases

as σ increases. From the formula (15), we can find that R^I increases as q decreases. As shown in Fig. 8 (c), when t1=t2=15, t3=100 and σ increases from 50 to 250, R^3 increases from about 0.19 to about 0.22.

2. Among the three parameters of t1, t2 and t3, t1 has the most influence on R^I, and t3 has the least influence on R^I. From formulas (14), (16), and (17), we can conclude that: t1, t2, t3 and CC form an exponential function with a base of less than 1. The characteristic of the exponential function indicates that the value of the function decreases significantly as the base decreases when *CC* is fixed. For example, as shown in formula (14), when *CC*=40, *m*=3500, *t1*=5, 10, 15, 20 and 25, $PGR^{1,3}$ is about 0.158, 0.291, 0.4, 0.497 and 0.577, respectively. Since $CC^i_{(r,c)} \approx \frac{CC}{L_2}$, the effect of parameter *t2* on R^I is much less than that of parameter t1 on R^I. Similarly, $1-\frac{t_3}{m\times L_3} > 1-\frac{t_2}{m}$, the effect of parameter t3 on R^I is much less than that of parameter t2 on R^I. For example, when t3=100, t1+t2=30, CC=20, σ=150 and t1 increases from 5 to 15, R^3 increases from about 0.06 to about 0.14; when t1=t2=15, t3=100, σ=250 and CC=10, 20, 30, 40 and 50, R^3 is about 0.07, 0.15, 0.22, 0.28 and 0.35, respectively.

C. Resilient Global connectivity

Global connectivity refers to the ratio of the number of nodes in the largest isolated part to the size of the whole network. In an isolated part, any two nodes can communicate with each other securely directly or indirectly. If the ratio equals to 95 percent, it means that 95 percent of the sensor nodes are connected and the remaining 5 percent are unreachable from the largest isolated component. So, the global connectivity metric indicates the percentage of nodes that are wasted because of their unreachability. For the global connectivity estimates, please refer to the literature [27].

From the analysis of section IV-B, it can be known that after *I* captures, previously secure communications may be compromised. Resilient global connectivity refers to the ratio of the number of nodes in the largest secure isolated part to the size of the whole network. In a secure isolated part, any two nodes can communicate with each other securely directly or indirectly after *I* captures. Zhao et al. have conducted a detailed analysis of the resilient global connectivity [21]-[22], in this paper, we only simulate it.

When t1=0, our scheme degenerates into schemes based on local key pools. However, in these schemes based on local key pools, once the size of deployment cells is determined, which are difficult to adapt to applications where the deployment error changes widely. As shown in Fig. 9 (a), in the first phase, when t1=0 and σ increases from 50 to 250, the resilient global connectivity decreases from about 0.993 to about 0.746. In our scheme, there are no common keys between these local key pools of different deployment phases, therefore, nodes, deployed in different phases, need to use pre-distribution keys of the global key pool to establish shared keys. When t1=0, nodes deployed in different phases cannot communicate securely, and the resilient global connectivity decreases significantly. As shown in Fig. 9 (a), when t1=0 and σ=250, the resilient global connectivity, from the 1st phase to the 2nd phase, and from the 2nd phase to the 3rd phase both decreased by about 0.15. When t1> 0, all nodes in the network can establish shared keys by using the t1 keys from the global key pool, so, the resilient global connectivity significantly increase and their values remain nearly unchanged as the deployment phase increases. As shown in Fig. 9 (a), when t1 = 5 and σ = 250, the resilient global connectivity of the 1st phase and the 3rd phase is about 0.996 and 0.993, respectively.

D. Comparison with the state-of-the-art technique

In this subsection, resilient local connectivity and resilient global connectivity of our scheme and SS-LM scheme is compared [26].

In [26], the division of the deployment area is the same as our scheme (see Fig. 1). Each cell is a basic cell. In order to enable nodes, whose deployment points are adjacent, to establish shared keys, the neighbor cells are divided into multi-layer contact cells according to the distance from the basic cell. The layer 0 contact cell is the basic cell itself. In [26], it has been proved that the upper limit of the number of contact cells of each basic cell is:

$$1+6\times\sum_{t=1}^{T} t \tag{18}$$

where T represents the total number of layers of the contact cells.

Each basic cell should generate shared keys for its contact cells. If the size of the key pool of each basic cell is m_B, then

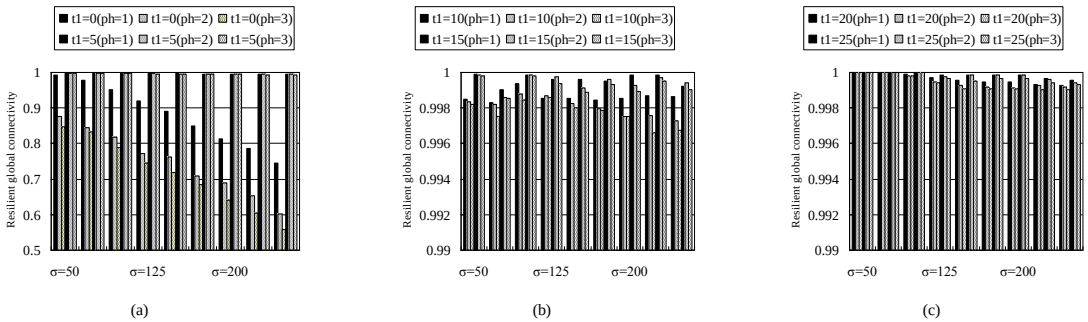
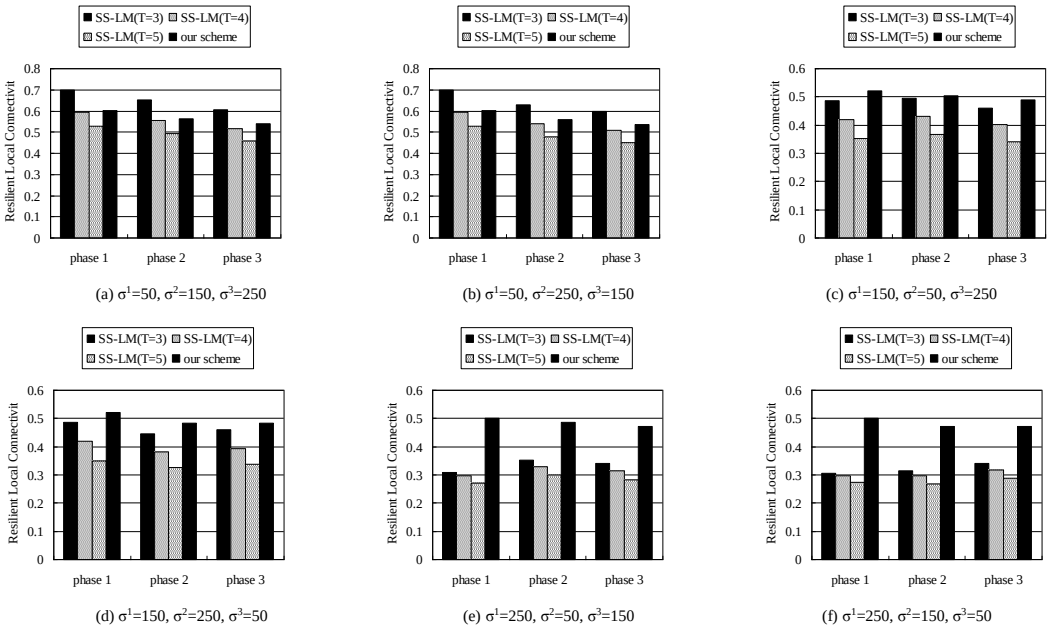


Fig. 9. Resilient global connectivity as a function parameters σ, t_1,t_2 and t_3. Parameter settings are the same as in Fig. 6.



(a) σ^1=50, σ^2=150, σ^3=250 (b) σ^1=50, σ^2=250, σ^3=150 (c) σ^1=150, σ^2=50, σ^3=250

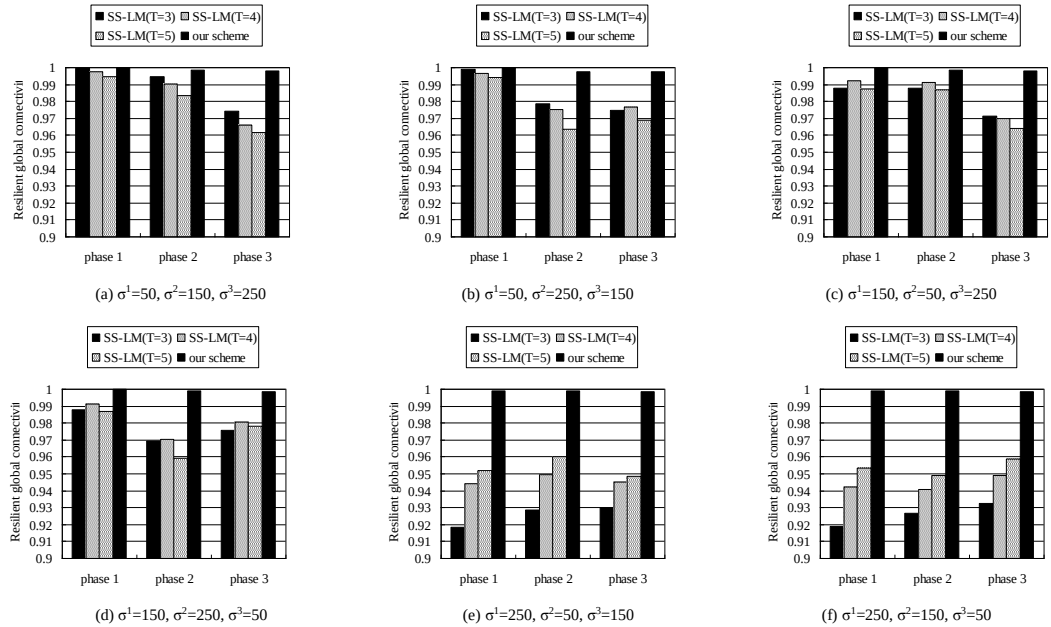
(d) σ^1=150, σ^2=250, σ^3=50 (e) σ^1=250, σ^2=50, σ^3=150 (f) σ^1=250, σ^2=150, σ^3=50

Fig. 10. Comparisons of resilient connectivity.

the upper limit of the size of the contact key pool of each basic cell, namely m_C, is:

$$m_C = m_B \times \left(1 + 6 \times \sum_{t=1}^{T} t\right) \tag{19}$$

In [26], in order to adapt to a large deployment error, the value of T must be increased, but from formula (19), it can be concluded that the size of the contact key pool will increase significantly with the increase of T. And when m_B is fixed, its connectivity will significantly decrease. As shown in Fig. 10 (a), when T increases from 3 to 5, RC^1 decreases from about 0.7 to about 0.53. However, as T increases, the performance of the scheme becomes more stable. From the comparisons between Fig. 10 (a) and Fig. 10 (f), in the two cases, it can be concluded that when T increases from 3 to 5, the decrement of RC^1 is about 0.39 and 0.26, respectively. From Fig. 10, it can also be derived: in either case, when T=3, the value of RC^I is higher than that when T=4 and T=5. However, when T=3, nodes within a basic cell cannot establish shared keys with nodes within the contact cells outside 3 layers. Hence, when the deployment error increases, it may lead to a decrease in resilient global connectivity (see Fig. 11). In [26], it is also not possible to adjust the T's value dynamically to adapt to the applications of different deployment errors. For example, in the 1st phase



(a) σ^1=50, σ^2=150, σ^3=250 (b) σ^1=50, σ^2=250, σ^3=150 (c) σ^1=150, σ^2=50, σ^3=250

(d) σ^1=150, σ^2=250, σ^3=50 (e) σ^1=250, σ^2=50, σ^3=150 (f) σ^1=250, σ^2=150, σ^3=50

Fig. 11. Comparisons of resilient global connectivity.

and the 2nd phase, when the deployment errors are 150 and 250, respectively, *T* should be equal to 4 and 5, respectively. However, in the 1st phase, there are no common keys between a basic cell and its 5th contact cells. In other words, in the 2nd phase, nodes within a basic cell cannot establish shared keys with nodes, deployed in the 1st phase, and within the 5th contact cells of the basic cell. Therefore, dynamically adjusting the value of *T* cannot achieve the purpose of maintaining stable performance of the network under different deployment errors. In addition, from Fig. 10, in SS-LM scheme, it can be concluded that the deployment error in the first stage has a greater impact on the resilient local connectivity. This is because after the 1st phase, it is assumed in subsequent phases that both nodes' captures and new nodes' addition happen within the same local scope, and fewer nodes are added, the impact of deployment errors on resilient local connectivity is reduced.

Fig. 11 shows the comparisons of the resilient global connectivity of the two schemes. From the deployment model, it can be seen that the probability that two nodes, whose deployment points far away from each other, become actual neighbors increases as the deployment error increases. In our scheme, the one-way associated key management model is adopted. When the deployment error is large, the number of keys distributed from the global key pool can be increased to increase the probability of nodes in the network establishing shared keys, thereby reducing the probability that nodes far away from the deployment point become isolated nodes because of not establishing shared keys with surrounding nodes. In SS-LM, once the value of the parameter *T* is determined, it is difficult to change it. When the set value of *T* is less than the actual deployment error requirement, it will increase the probability that nodes far away from the deployment point become isolated nodes because of not establishing shared keys with surrounding nodes, resulting in a significant decrease in the global connectivity. From the comparison results, we can conclude that the resilient global connectivity of our scheme is significantly better than that of the SS-LM scheme. In SS-LM, when the deployment error is large, the global connectivity can be improved by increasing *T*. As shown in Figs. 11 (e) and 11 (f), when $\sigma 1 = 250$ and $T=3$, 4, and 5, the resilient global connectivity is about 0.92, 0.94, and 0.95, respectively.

V. Conclusion

The network performance of existing schemes based on deployment knowledge changes dramatically as the deployment error changes. In this paper, we proposed a new key management scheme based on one-way associated key management model. In this scheme, the key pool consists of the global key pool layer and the local key pool layer, and keys from the local key pool layer can be calculated by using keys from the global key pool layer. In applications where the deployment error varies widely, this scheme can maintain stable network performance by dynamically adjusting the number of keys pre-distributed from the global key pool and local key pools. Detailed analysis and numerical simulation indicate that: in the case of large variation of the deployment error, the dynamic adjustment of parameters can make the resilient global connectivity of our scheme reach about 1. Under the same conditions, the resilient global connectivity of the related scheme changes greatly, in the worst case, it is about 0.92, which means that about 8% of nodes are wasted.

References

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

[2] K. Piotrowski, P. Langendoerfer, S. Peter, "How Public Key Cryptography Influences Wireless Sensor Node Lifetime," in: ACM SASN 2006.

[3] 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.

[4] F. Gandino, R. Ferrero, B. Montrucchio, et al. "Fast Hierarchical Key Management Scheme With Transitory Master Key for Wireless Sensor Networks," IEEE Internet of Things Journal, vol. 3, no. 6, 1334-1345, 2016.

[5] S. Li, B. Zhou, J. Dai, et al. "A secure scheme of continuity based on two-dimensional backward hash key chains for sensor networks," IEEE Wireless Communications Letters, vol. 1, no. 5, pp. 416-419, 2012.

[6] A. K. Das, "An efficient random key distribution scheme for large-scale distributed sensor networks," Security and Comm. Networks, vol. 4, no. 2, pp. 162-180, 2011.

[7] S. Li, B. Zhou, J. Dai, et al., "A Secure Scheme of Continuity Based on Two-Dimensional Backward Hash Key Chains for Sensor Networks," IEEE Wireless Communications Letters, vol. 1, no. 5, pp. 416-419, 2012.

[8] B. Zhou, S. Li, J. Wang, et al., "A pairwise key establishment scheme for multiple deployment sensor networks," International Journal of Network Security, vol. 16, no. 3, pp. 221-228, 2014.

[9] B. Zhou, J. Wang, S. Li, et al., "A Secure Scheme of Continuity in Static Heterogeneous Sensor Networks," IEEE Communications Letters, vol. 17, no. 9, pp. 1868-1871, 2013.

[10] S. Li, W. Wang, B. Zhou, et al., "A Secure Scheme for Heterogeneous Sensor Networks," IEEE Wireless Communications Letters, vol. 6, no. 2 , pp. 182-185, 2017.

[11] S. Li, W. Wang, B. Zhou, et al., "A (M,m) Authentication Scheme against mobile sink replicated Attack in Unattended Sensor Networks," IEEE Wireless Communications Letters, vol. 7, no. 2, pp. 250-253, 2018.

[12] B. Zhou , S. Li, W. Wang , et al., "An Efficient Authentication Scheme Based on Deployment Knowledge Against Mobile Sink Replication Attack in UWSNs," IEEE Internet of Things Journal, vol. 6, no. 6, 9738-9747, 2019.

[13] L. Li, G. Xu , L. Jiao, et al. "A Secure Random Key Distribution Scheme Against Node Replication Attacks in Industrial Wireless Sensor Systems," IEEE Transactions on Industrial Informatics, vol. 16, no. 3, pp.2091-2101, 2020.

[14] W. Du, J. Deng, Y. S. Han, et al., "A Key Predistribution Scheme for Sensor Networks Using Deployment Knowledge," IEEE Trans. Dependable and Secure Computing, vol. 3, no. 1, pp. 62-77, 2006.

[15] Z. Yu, Y. Guan, "A Key Management Scheme Using Deployment Knowledge for Wireless Sensor Networks," IEEE Trans. on Parallel and Distributed Systems, vol. 19, no. 10 pp. 1411-1425, 2008.

[16] A. Fanian, M. Berenjkoub, H. Saidi, and T. A. Gulliver, "A high performance and intrinsically secure key establishment protocol for wireless sensor networks," Computer Networks, vol. 55, no. 7, pp. 1849–1863, 2011.

[17] B. Zhou, S. Li, Q. Li, X. Sun, and X. Wang, "An Efficient and Scalable Pairwise Key Pre-distribution Scheme for Sensor Networks Using Deployment Knowledge," Computer Communications, vol. 32, no. 1, pp. 124-133, 2009.

[18] B. Zhou, J. Wang, S. Li, W. Wang, "A new key predistribution scheme for multi-phase sensor networks based on a new deployment mode," Journal of sensors, Article ID 573913, 10 pages, 2014.

[19] B. Zhou, J. Wang, S. Li, et al., "A Secure Scheme Based on Layer Model in Multi-Phase Sensor Networks," IEEE Communications Letters, vol. 20, no. 7, pp. 1421-1424, 2016.
[20] W. Gu, S. Chellappan, X. Bai, et al. "Scaling Laws of Key Predistribution Protocols in Wireless Sensor Networks," IEEE Transactions on Information Forensics and Security, vol. 6, no. 4, 2011, pp. 1370 – 1381.
[21] J. Zhao, "On Resilience and Connectivity of Secure Wireless Sensor Networks Under Node Capture Attacks," IEEE Transactions on Information Forensics and Security, 12(3): 557-571, 2017.
[22] J. Zhao, "Probabilistic Key Predistribution in Mobile Networks Resilient to Node-Capture Attacks," *IEEE Transactions on Information Theory*, 63(10): 6714-6734, 2017.

Sujun Li received the Ph.D. degree in computer science from Central South University, Changsha, China, in 2018.
She is currently a Vice Professor with the School of Information Science and Engineering, Shaoguan University, Shaoguan, China, and also with MOE-LCSM, School of Mathematics and Statistics, Hunan Normal University, Changsha, Hunan 410081, China. Her current research interests include sensor networks and information security.

Boqing Zhou received the Ph.D. degree in computer science from Hunan University, Changsha, China, in 2011.
He is currently a Vice Professor with the School of Information Science and Engineering, Shaoguan University, Shaoguan, China, and also with MOE-LCSM, School of Mathematics and Statistics, Hunan Normal University, Changsha, Hunan 410081, China. His current research interests include sensor networks and information security.

Jianxin Wang (SM'12) received the Ph.D. degree in computer science from Central South University, Changsha, China, in 2006.
He is currently a Professor with the School of computer Science and Engineering, Central South University. He has published over 100 papers in various international journals and refereed conferences. His current research interests include algorithm analysis and optimization, computer network, and bioinformatics.

Jingguo Dai is currently a Professor with the School of Information Science and Engineering, Shaoguan University, Shaoguan, China. His current research interests include sensor networks and information security.

Weiping Wang (M'13) received the Ph.D. degree in computer science from Central South University, Changsha, China, in 2004.
She is currently a Professor with the School of computer Science and Engineering, Central South University. Her current research interests include network optimization, network encoding, network security, and anonymous communication.

Huiyong Yuan is currently a Professor with the School of Information Science and Engineering, Shaoguan University, Shaoguan, China. His current research interests include sensor networks and information security.

Yun Cheng received the Ph.D. degree in computer science from the National University of Defense Technology, Changsha, China, in 2001.
He is currently a Professor with the Department of Information, Hunan Institute of Humanities, Science and Technology, Loudi, China. His current research interests include algorithm analysis and computer network.

Jie Wu (F'09) received the B.S. degree in computer engineering and the M.S. degree in computer science from the Shanghai University of Science and Technology, Shanghai, China, in 1982 and 1985, respectively, and the Ph.D. degree in computer engineering from Florida Atlantic University, Boca Raton, FL, USA, in 1989.

He is the Chair and a Laura H. Carnell Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. He was a Program Director with the National Science Foundation, Alexandria, VA, USA, and a Distinguished Professor with Florida Atlantic University, Boca Raton, FL, USA. He regularly publishes in scholarly journals, conference proceedings, and books. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu was a recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award. He serves on several editorial boards, including the IEEE TRANSACTIONS ON SERVICE COMPUTING and the *Journal of Parallel and Distributed Computing*. He was the General Co-Chair/Chair of the IEEE MASS 2006, the IEEE IPDPS 2008, the IEEE ICDCS 2013, and ACM MobiHoc 2014, as well as the Program Co-Chair of the IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, an ACM Distinguished Speaker, and the Chair of the IEEE Technical Committee on Distributed Processing. He is a CCF Distinguished Speaker.