This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3524409

A new pairwise key scheme based on deployment knowledge in multi-phase sensor networks

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

Abstract—The lifetime of a sensor network is generally longer than that of a single sensor node, so to ensure the connectivity of the network, new nodes need to be deployed into the network in multi-phase. Such networks are called multi-phase sensor networks. In sensor networks, establishing a pairwise key between two neighboring nodes is a fundamental security requirement. However, for sensor nodes with strictly limited capabilities, there are great challenges in establishing a pairwise key between two neighboring sensor nodes deployed at different phases. In order to solve this problem, some scholars have proposed some solutions based on backward key-chains and forward key-chains, however, how to further improve the networks' resilience against node capture attacks still needs further research; and some scholars have proposed some solutions based on key pools of each phase are independent, but how to improve the probability of establishing a pairwise key between two neighboring nodes deployed at different phases is still a very challenging problem. In this paper, we propose a new pairwise key scheme based on deployment knowledge. Theoretical analysis and simulation show that the proposed model can achieve high probability of establishing a pairwise key between two neighboring nodes and good resilience against node capture attacks by setting appropriate parameters.

Index Terms—*multi-phase sensor networks, pairwise key, key update.*

I. Introduction

Sensor networks (SNs) usually consist of a large number of sensor nodes. A sensor node is battery powered and equipped with integrated sensors, data processing, and short-range radio communication capability. So, the lifetime of a single sensor node is much shorter than that the operating lifetime of the network. To ensure the connectivity of a network, new nodes need to be periodically deployed into the network. Such networks are called multi-phase sensor networks (MSNs). MSNs are often deployed outdoors or even in hostile environments, and are vulnerable to various attacks [1]-[2]. Hence, it is important to protect communication among sensor nodes to maintain message confidentiality and integrity. As one of the most fundamental security services, pairwise key establishment enables two neighboring sensor nodes to communicate securely with each other. Therefore, different key pre-distribution schemes have been developed for SNs [3]-[15]. However, in these schemes [3]-[15], throughout the entire lifecycle of the network, the same key pool is used. Obviously, they are not suitable for MSNs. This is because when an adversary captures a certain number of nodes, the keys in the key pool will basically be compromised. This means that newly deployed nodes not only do not help the network, but also pose a greater threat. Because the key information pre-distributed to them has been compromised.

A. Motivation

To enable nodes deployed at different phases to establish pairwise keys, some scholars construct multi-phase key pool using one-way hash function [11]-[20]. Multi-phase key pools include the following two categories: one is backward multi-phase key pools, and the other is forward multi-phase key pools. So far, the backward multi-phase key pools are generally constructed by using one-dimension backward key chains [11] - [13], two-dimension backward key chains [14] and three-dimension backward key chains [15], [16]. For a backward key chain, when the generation key of a certain phase is compromised, the keys generated in these phases which are less than or equal to the phase are no longer secure. In [17]-[20], some solutions have been proposed to improve the security of networks by adding forward multi-phase key pools. Similarly, for a forward key chain, if the generation key of a certain phase is compromised, these keys generated in these phases which are equal to or greater than the phase are no longer secure. That is, the adversaries can obtain a large number of keys of a forward key pool by compromising a small number of sensor nodes. When most keys in the forward key pools are compromised, these key pools will be ineffective. To improve the resilience against node capture attacks of networks, some schemes have been proposed in [21-23] by using independent key pools at each phase. The focus of such schemes is how to establish a pairwise key between two neighboring nodes deployed at

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: chy8370002@gmail.com.

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

different phases. In [21], Durresi et al. proposed a scheme in which two neighboring nodes deployed at different phases can establish a pairwise key with the help of bridge nodes. Obviously, if a large number of bridge nodes are compromised or unable to function properly, these nodes deployed at different phases will not be able to communicate securely with each other. In [22], Zhou et al. proposed a scheme where a pairwise key can be established directly between two neighboring nodes deployed at two adjacent phases. In this scheme, as the number of deployment phase increases, its connectivity decreases rapidly. In [23], during a node's lifetime, Ergun and Savas proposed a scheme where a node can probabilistically establish a pair-wise key with its neighbors using the pre-distribution keys. Compared with the scheme in [22], the connectivity of the scheme declines slower. However, the storage overhead of the scheme is high, and the resilience against node capture attacks also declines. Therefore, further research is needed to address the issue of secure connectivity between two neighboring nodes deployed at different phases.

B. Main contribution of our scheme

In this paper, by using deployment knowledge, we propose a new pairwise key scheme. Our main contributions are as follows:

1) In our scheme, the key pool of each deployment phase is independent, but nodes only need to store keys from one phase, and communication between nodes deployed at different phases is achieved through key update methods. This reduces the storage overhead of nodes, and at the same time, this scheme uses a combination of key space and master key technology to improve the resilience against node capture attacks.

2) A new method for establishing path keys has been proposed. The essential difference between this method and traditional methods is that it does not require two adjacent nodes on the path key to be actual neighboring nodes. Under the same conditions, it increases the probability of nodes establishing a shared key through path keys.

3) In our scheme, using deployment knowledge, the shared keys establishment and the keys update can be completed locally.

C. Organization

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

II. Related work

The first random key pre-distribution scheme, namely E-G scheme, was proposed by Eschenauer and Gligor [3], in which each sensor selects a set of keys randomly from a large key pool before deployment to establish a shared key between two nodes. To enhance the security of E-G scheme against small-scale attacks, q-composite scheme was proposed [4], in which q common keys are required for two nodes to establish a shared key. Liu and Ling proposed a random key pre-distribution scheme where the key pool is constructed by using the key space [5]. Du et al. also proposed a similar scheme based on symmetric matrix [6]. This scheme exhibits a nice threshold property: when the number of nodes being compromised is less than the threshold, the probability of communications between any two additional nodes being compromised is close to zero. Msolli et al. proposed a new method for constructing a key pool [7], in which the key pool consists of the keys and these keys' hashed value. Due to the fact that the adversary cannot obtain the original key after obtaining its hashed key, its security performance is improved as compared to the E-G scheme [3]. Gandino et al. proposed a method to reduce energy consumption during the process of establishing shared keys [8], which randomly selects a portion of keys from multiple shared keys for the establishment of shared keys. Gandino et al. proposed a method for establishing shared keys between nodes based on master keys [9]. Two neighboring nodes in the network can use the master key to establish a shared key. Once the node completes the key establishment, the master key is deleted. But if the master key is known by the adversary before deletion, the entire network is no longer secure. This article proposes a new method to reduce the time required for nodes to establish shared keys. Altun et al. proposed a method for establishing shared keys between nodes based on the superior property of wireless channels [10].

In order to make the proposed key management schemes applicable to MSNs, some scholars have proposed methods based on online key distribution and updates [11] - [12]. In these methods, a small number of keys are pre-distributed to sensor nodes before deployment, but authentication between nodes and keys distribution require the participation of the base station. Khah et al. proposed a multi-level security management scheme [13], in which the closer the base station is, the higher the security level of the network. At the same time, with the participation of the base station, this scheme can achieve the generation, distribution, and updating of multiple kinds of keys. Das proposed a two-phase key pool scheme [14]. The nodes deployed at the first phase select keys from these two key pools, but the keys selected from the second key pool need to be hashed before pre-distribution. New nodes added to the network require storing keys from the second key pool. The two-phase key pool improves the security performance of using only one key pool. Zhou et al. and Messai constructed multi-phase key pools by using one-dimension backward key chains [15-17]. In [15], nodes deployed at the ith phase, which are pre-distributed keys from the key pool of the ith phase only. Usually, most of the pre-distribution keys need to be hashed before being stored in nodes. It is computationally infeasible to infer the original keys after processed by hash operation. Therefore, the resilience of this schemes is high, but there is a loss in local connectivity. Similar to the scheme in [15], in [16], a small number of keys are pre-distributed to a sensor node before deployment, and a large number of keys after being processed by hash operation, are pre-distributed to a mobile sink. In the POK scheme [17], the key server pre-distributes a key chain to each pair of nodes deployed in the first phase. This leads to a significant decrease in the probability of nodes being able to directly establish pairwise keys as the network size expands. In [18], Li et al. built multi-phase backward key pools

using two-dimension backward key chains. This solution has good resilience when a small number of nodes are compromised during the pairwise key establishment phase. In [19], to resist mobile sink replication attacks, Zhou et al. construct multi-phase key pools by using 3-dimensional backward key chains. In [20], Li et al. proposed a kind of three-dimension backward key chain technology based on deployment knowledge. By adjusting the number of pre-distribution keys of the first, second and third dimension, the scheme can be applied to various deployment scenarios. To improve the networks' resilience against node capture attacks, in [21], Castelluccia and Spognardi proposed a solution in which multi-phase key pools are constructed by using backward key chains and forward key chains simultaneously. In the solution, nodes deployed in the ith phase, which need to be pre-distributed the same number of keys from the ith phase forward key pool and the ith phase backward key pool, respectively. However, when a small number of nodes are compromised, the whole forward multi-phase key pool will be ineffective. That is, this scheme cannot significantly improve the network resilience against node capture attacks. In [22], Ito et al. proposed a solution, which can improve the resilience of the forward multi-phase key pool by using key space technology. But when a large number of nodes are compromised, the resilience against node capture attack is degraded to the scheme based on key space in [5]. In [23], Sarimurat and Levi introduced a new key update solution. In this solution, the jth key of the i+1th phase can be calculated jointly by using the jth and the j+1th keys of the ith phase. Therefore, a new key will be added to the key pool when a key update occurs. As compared with the scheme in [23], its connectivity decreases, but its resilience improves. In [24], Messai and Seba proposed a scheme, where IDs of keys can be updated. At the same time, to improve the scheme's resilience, a new method of establishing pairwise key between nodes based on 2-composite scheme was proposed.

Durresci et al. proposed the SCON scheme [25]. In this scheme, key pools of each phase are independent, and nodes deployed in the ith phase store keys from the ith phase key pool only. To enable the nodes deployed at the two adjacent phases to establish pairwise keys, bridge nodes are introduced. The bridge nodes deployed in the ith phase store keys from the ith phase key pools and the i-1th phase key pools simultaneously. Therefore, the bridge nodes of the ith phase can probabilistically establish pairwise keys with nodes whose deployment phases are adjacent to them. In the scheme, bridge nodes are difficult to be deployed, and its resilience against bridge node capture attacks is poor. In order to improve the applicability of the SCON scheme, Zhou et al. constructed key pools using deployment knowledge and one-dimension key chains, which makes key pools of two neighboring deployment phases are associated [26]. However, only nodes deployed in the two adjacent deployment phases can directly establish pairwise keys. In [27], to improve the local connectivity, Ergun and Savas proposed the RGM scheme. In the RGM scheme, before deployment, nodes deployed in the ith phase store keys from the key pools of the ith phase to the (i+Gw-1)th phase, where Gw represents a generation window which is the time period where this node is alive. Obviously, compared with [25] and [26], its local connectivity is improved significantly. However, its resilience declines rapidly because of carrying a lot of key information.

III. DEPLOYMENT KNOWLEDGE AND THREAT MODELS

Here, we will introduce our scheme from the following four aspects: 1. deployment model and threat model. 2. keys pre-distribution. 3. pairwise key establishment; 4. key update.

A. Notations

In this paper, we use the following notations for the description convenience.

TABLE I NOTATIONS USED

Notation	Description
$S^i_{(r,c)}$	The set of nodes being deployed in the cell (r,c) at the ith phase
Gw	A generation window which is the time period where a sensor node is alive.
t	The number of nodes in $S^i_{(r,c)}$, which equals to $\left\|S^i_{(r,c)}\right\|$
R	The communication radius of sensor nodes
Avg	The average number of neighbor nodes across the entire deployment region. The setting of this parameter value directly affects the performance of the scheme.
Len	The length of regular hexagons
N	The maximum phase of nodes deployed to the network
P	The probability of nodes in $S^i_{(r,c)}$ locating in (r,c) and $NC_{(r,c)}$. The setting of this parameter value will have a significant impact on the performance of the solution.
H()	A one-way hash function
$H_K()$	A one-way hash function with the key *K*
$E_K()$	A encryption function with the key *K*
min(x,y)	Represents the minimum value of x and y
max(x,y)	Represents the maximum value of x and y
P_{NC}	Represents the probability that a node in $S^i_{(r,c)}$ selects nodes from the set of neighbor cells deployed in the cell (r,c) recently. This parameter is important for the performance of this scheme.
P_{SC}	Represents the probability that a node in $S^{i,j}_{(r,c)}$ selects nodes from $S^{i+1}_{(r,c)}$. This parameter is an important parameter for the performance of this scheme.
CC^i	The number of nodes captured by adversaries before the key update of newly deployed nodes at the ith phase is completed. If CC^i is equal in all phases, use CC instead. This parameter is important for the performance of the scheme.
%	Represents the residual operator
$\oplus$	Represents the XOR operation

B. Deployment Knowledge Model

Some scholars have proposed deployment models for randomly distributed sensor nodes in networks [3] - [8], which have the advantage of supporting node mobility. To adapt to this model, the entire network shares a global key pool. Nodes only need to store keys selected from the global key pool before deployment. However, in the application of static sensor

networks, nodes are highly likely to fall around the deployment point [28] - [32]. Therefore, scholars divide the deployment area into cells, and at the same time, the global key pool is divided into multiple key pools based on deployment cells. The key pools of two non-neighboring cells do not have common keys. Before deployment, nodes store keys from the key pool of the corresponding deployment cell. This model is called a model based on deployment knowledge. It is obvious that the solutions based on deploying knowledge can not only reduce the storage overhead of nodes, but also improve the probability of establishing shared keys between nodes and the resilience.

In our scheme, the deployment model is similar to the scheme in [23], the difference is that the nodes of our scheme are deployed by multiple phases. The set of nodes deployed in the ith phase is represented by S^i. A target field is partitioned into hexagon cells (see Fig. 1). Each cell has a deployment point that resides in the center of the cell. The set of nodes deployed in the cell at the ith phase is represented by $S^i_{(r,c)}$. Node distribution follows two-dimensional Gaussian distributions with the deployment point (x', y') as center, as follows:

$$f(x,y)=\frac{1}{2\pi\sigma^2}e^{-\left[(x-x')^2+(y-y')^2\right]/2\sigma^2} \tag{1}$$

where σ is the standard deviation of distribution. Combined with the formula (1), the length of the cell *Len* can be estimated using the following formula:

$$Len=\frac{2\sigma\sqrt{-2\ln(1-P)}}{3\sqrt{3}} \tag{2}$$

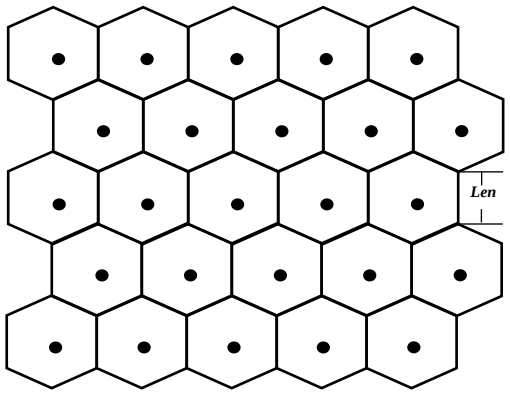


Fig. 1 Deployment area division diagram, • represents the deployment point.

C. Architecture of the Network

The network consists of a large number of sensor nodes and a base station. After the network is deployed, the sensor nodes in the network are stationary. Nodes in the network transmit data to the base station through one hop or multiple hops. The base station can be located at any position in the network, responsible for pre-distributing key information to nodes, receiving and analyzing data sent by nodes in the network.

D. Threat model

In our scheme, sensor nodes can be compromised. That is to say, if an attacker captures a sensor node, all key information it holds will also be compromised. Moreover, the adversary may pool the keying materials from multiple compromised nodes to break the security of the network or to launch advanced attacks. such as eavesdropping, false data injection attacks, etc.

In our scheme, because the time of pairwise key establishment and update is short, we assume that in the two phases only a few nodes of newly deployed are compromised [2], [8], [16], [17], [20], [21]. Meanwhile, we assume that the base station is secure.

E. Keys pre-distribution

Keys pre-distribution is divided into the following three steps. For the convenience of description, the following will take the node a ($a \in S^i_{(r,c)}$) as an example for detailed introduction.

Step 1. The pre-distribution keys information for pair-wise key establishment between two neighboring nodes deployed in the same set.

The key server generates a t-degree binary symmetric variable polynomial for the set $S^i_{(r,c)}$ [29]:

$$K^i_{(r,c)}(x,y)=\sum_{j_1,j_2=0}^{t}A^i_{j_1,j_2}x^{j_1}y^{j_2} \tag{3}$$

where $A^i_{j_1,j_2}=A^i_{j_2,j_1}$ and $t=\left|S^i_{(r,c)}\right|$.

The pre-distribution key $K^i_{(r,c)}(x,y)$ for node a can be calculated by using the following formula:

$$K^i_{(r,c)}(ID_a,y)=\sum_{j=0}^{t}B_j y^j \tag{4}$$

Step 2. The pre-distribution keys information for pairwise key establishment between two nodes deployed in two neighboring cells.

Supposing the cell (r_1, c_1) is the neighbor cell of the cell (r,c). $S^{i_1}_{(r_1,c_1)}$ represents that nodes deployed in the cell (r_1, c_1) recently when nodes in the set $S^i_{(r,c)}$ are deployed. Node a ($a \in S^i_{(r,c)}$) selects t_1 horizontal connection nodes from the $S^{i_1}_{(r_1,c_1)}$ according to the formula (5).

$$ID_b=ID_{b_1}+H^L\left(ID_a \| r_1 \| c_1 \| i_1\right)\%t \tag{5}$$

where ID_{b_1} represents the minimum ID in the set $S^{i_1}_{(r_1,c_1)}$. $H\left(H^L(\inf)\right)=H^2\left(H^{L-1}(\inf)\right)=\ldots=H^{L+1}(\inf)$. If ID_b has been assigned to itself, it needs to be recalculated until the calculated ID_b is not pre-assigned to itself (The detailed process can be found in Algorithm 1).

$$t_1=\lfloor P_{NC}\cdot t\rfloor \tag{6}$$

Assuming node b ($b \in S^{i_1}_{(r_1,c_1)}$) is the horizontal connection node selected by the node a, the following key will be pre-distributed to it:

$$K^{i_1}_{(r_1,c_1)}(ID_b,ID_a) \tag{7}$$

Step 3. The pre-distribution keys information for pairwise key establishment between nodes deployed at different phases.

These keys consist of the following two parts. The first part is the pre-distribution master key MK_a. Below is a detailed introduction to the second part of these keys.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3524409

Algorithm 1: Algorithm for calculating identity without repetitive.

```
void computing_non_rep_ID(int ID0, int ID1, ID_List list[])
//ID0 represents the ID_a, ID1 represents the ID_{b_1}
{ int f[t],i,h0;
  for(i=0;i<t;i++) f[i]=0;
  h0= H(ID0||r1||c1||i1)%t ;
  for(i=0; i<t1; )
  {
    if(f[h0]==0) { f[h0]=1; list[i]=ID1+h0; i++; }
    h0 = H(h0)%t ;
  }
}
```

Node a ($a \in S_{(r,c)}^{i}$) will select t_2 vertical connection nodes from the set $S_{(r,c)}^{i+1}$ according to the formula (5).

$$t_2 = \lfloor P_{SC} \cdot t \rfloor \tag{8}$$

Assuming node b ($b \in S_{(r,c)}^{i+1}$) is the vertical connection node selected by the node a, then the following keys will be pre-distributed to the node a:

$$K_{a,b} = \begin{cases} H\left(H_{MK_b}(ID_a)\right) & i=1 \\ H_{MK_b}(ID_a) & i>1 \end{cases} \tag{9}$$

where MK_b represents the master key of the node b.

F. Pairwise key establishment

After all nodes are deployed, the shared key establishment phase will immediately start, and the establishment process is as follows:

Step 1. Node broadcasts its own ID;

Step 2. The process of directly establishing a shared pair-wise key between two neighboring nodes a and b includes the following three situations:

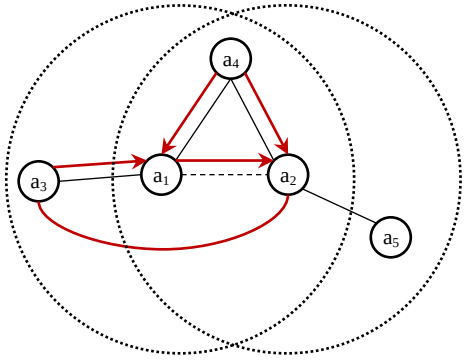


Fig. 2 Path key establishment process. A black solid line indicate that two adjacent nodes can directly establish a pairwise key, a black dotted line represent that two adjacent nodes cannot directly establish a pairwise key, a red solid line indicate that two non-adjacent nodes can directly establish a pairwise key, and a red solid line with direction represent the direction of the path key transmission.

Case 1. If a ($a \in S_{(r,c)}^{i}$) and b ($b \in S_{(r,c)}^{i}$), the pairwise key between them is: $K_{(r,c)}^{i}(ID_a, ID_b)$;

Case 2. If a ($a \in S_{(r,c)}^{i}$) and b ($b \in S_{(r_1,c_1)}^{i_1}$), the node b can calculate the key $K_{(r_1,c_1)}^{i_1}(ID_b, ID_a)$ pre-distributed to the node a;

Case 3. If a ($a \in S_{(r,c)}^{i}$) and b ($b \in S_{(r,c)}^{i+1}$), the node b can calculate the pairwise key between them using the formula (9);

Step 3. If two neighboring nodes a1 and a2 cannot directly establish a pairwise key, the path key establishment process is initiated. The process of establishing a path key is described as follows:

Step 3-1. The node broadcasts the ID of the communication partner to its neighboring nodes;

Step 3-2. After receiving the above information, neighboring nodes establish path keys according to the following two situations (The detailed process can be found in Algorithm 2):

Algorithm 2: Algorithm for establishing path keys between neighboring nodes

```
void Path_key(int ID0, int ID1, int ID2, Neig_List nlist[], Share_Flag sf)
/*ID0 represents the ID_{a_1}, ID1 represents the ID_{a_2}, nlist[] represents the
neighbor nodes list of ID2, Sf is a variable used to identify that ID2 can
directly establish a shared key with ID0 and ID1 simultaneously. If it
can, the value is 1, otherwise it is 0. If sf is 1, then sk1 and sk2 are used
to represent the shared key between ID2 and ID0 and ID1, respectively.
*/
{ int p1,p2,rn;
  if (sf==1)
  { p1=find(nlist,ID0);
  /*find is an algorithm for searching for elements in a list. If found, it
  returns the position of the corresponding element in the list. If not
  found, it returns -1. */
    p2=find(nlist,ID1);
    rn=random();
    /*random() represents a function that generates random numbers*/
    if (p1!=-1 && p2!=-1)
    { send(ID0||ID2||E_Sk1(rn)||H_Sk1(rn));
  //send() represents a function for sending information
      send(ID1||ID2||E_Sk2(rn)||H_Sk2(rn));
    }
    if (p1!=-1 && p2==-1)
    {
  send(ID0||ID2||ID1||E_Sk1(rn)||H_Sk1(rn)||E_Sk2(rn)||H_Sk2(rn));
      }
    if (p1==-1 && p2!=-1)
    {
  send(ID1||ID2||ID0||E_Sk2(rn)||H_Sk2(rn)||E_Sk1(rn)||H_Sk1(rn));
      }
  }
}
```

Case 1. If this node can directly establish pairwise keys with a1 and a2 and belongs to their common neighbor nodes, then, this node generates a random number as the path key for a1 and a2 and securely sends this key to them. As shown in Fig. 2, the node a4;

Case 2. If this node is not a1 and a2's common neighbor node, but can directly establish pairwise keys with them

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3524409

respectively, as shown in Fig. 2, the node a3, then, the node a3 generates a random number as the path key for a1 and a2, and sends two encrypted data of the path key to a1, a1 receives its own encrypted data and then forwards the other encrypted data belonging to a2 to it. After decryption, a1 and a2 can obtain their shared path key.

Step 3. If there are multiple path keys k_1, k_2,..., k_n, the shared key between them is: $sk = k_1 \oplus \cdots \oplus k_n$ 。

G. Key Update

The key pre-distribution method shows that if two nodes with adjacent or identical deployment points, and their deployment phases are not adjacent, it is impossible to directly establish a pairwise key between them. Therefore, in order to enable two nodes whose deployment phases are non-adjacent to establish a pairwise with a certain probability, in this paper, a key update method is proposed. For the convenience of description, the following will take the newly deployed node $S_{(r,c)}^{i}$ as an example to explain the key update process:

Step 1. Build a cluster around the newly deployed nodes.

Step 1-1. Newly deployed nodes compete to become cluster heads according to the following rules：

Assuming $S_{(r,c)}^{i}$ is the newly deployed nodes set, NN_b is the neighbors set of node b ($NN_b \subset S_{(r,c)}^{i}$). When the following conditions are met:

$$\left|NN_b\right| \geq \left(1 - e^{-\frac{R^2}{2\sigma^2}}\right) \times \left|S_{(r,c)}^{i}\right| - \Delta \cdot T \tag{10}$$

The node b becomes the alternative cluster head, and the greater the value of $\left|NN_b\right|$, the higher the probability of becoming the cluster head. In the formula (10), Δ represents the error value, T represents the number of repetitions and its initial value is 1. If no nodes in $S_{(r,c)}^{i}$ meet the conditions of the formula (10), T increases by 1 successively until there exist nodes meet the conditions of the formula (10). For the convenience of description, the selected cluster head is represented by $CS_{(r,c)}^{i}$. $CS_{(r,c)}^{i}$ broadcast the following clustered messages:

$\inf_C = \left\{ ID_{CS_{(r,c)}^{i}}, h \right\}$, where h=0.

Step 1-2. When node d receives $\inf_C$, if d belongs to one of the following two types of nodes, it needs to participate in the broadcast:

Type 1. $d \in S_{(r,c)}^{i'}$ $(1 \leq i' \leq i)$;

Type 2. the neighbors of node d contain the first type of nodes those have not received the $\inf_C$. As shown in Fig. 3, there are five nodes: 7, 12, 15, 20, and 25.

If d participates in broadcasting, assuming h' is the smallest h value received by d, then d broadcasts the following message:

$\inf_C = \left\{ ID_{CS_{(r,c)}^{i}}, h'+1 \right\}$

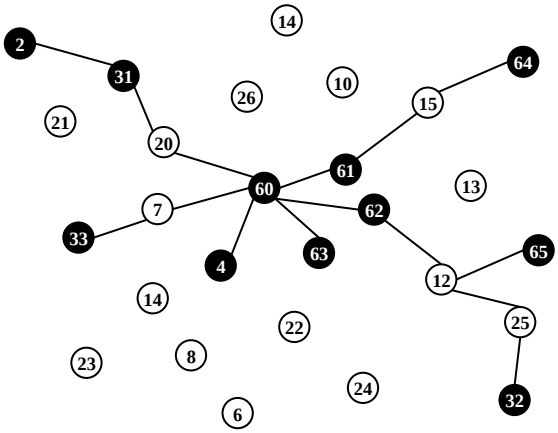


Fig. 3 Schematic of nodes involved in broadcasting during clustering. In the figure, nodes with a node number greater than or equal to 60 are newly deployed nodes, and the black points represent nodes deployed in the same cell with these newly deployed nodes. The cluster head is the node 60.

Step 2. Nodes deployed in the cell (r,c), which need to send their IDs to $CS_{(r,c)}^{i}$.

Step 3. the cluster head fuses these IDs information into a list of nodes, finally, $CS_{(r,c)}^{i}$ sends the list to newly deployed nodes in the cluster.

Step 4. The nodes in $S_{(r,c)}^{i}$ calculate the nodes which need to be updated and their update keys, and securely send these update keys to the corresponding nodes. Taking a2 as an example, we will provide a detailed introduction.

As shown in the Fig. 4, assuming b_3 and b_4 are vertical connection nodes selected by a_2. When nodes in the set $S_{(0,0)}^{2}$ are deployed, b3 and b4 are responsible for updating the key of node a2. Taking b3 as an example, we will explain the calculation process of keys update.

Assuming c_3 and c_1 are vertical connection nodes selected by b_3, c_3 and c_2 are vertical connection nodes selected by b_4. Then a_2 selects the first update path as follows: $a_2 \leftrightarrow b_3 \leftrightarrow c_3$. Since c_3 has been selected as the node for the first path, the second path can only be: $a_2 \leftrightarrow b_4 \leftrightarrow c_2$. From the formula (9), it can be concluded that b_3 only require a simple hash operation to calculate the pairwise key with a_2. Meanwhile, K_{b_3,c_3} is pre-distributd to b_3. After K_{b_3,c_3} performing the following hash processing, it will be used as the pairwise key for a_2 and c_3:

$$K_{a,c_3} = H_{k_{b_3,c_3}}\left(ID_a\right) \tag{11}$$

Then K_{b_3,c_3} will be safely transmitted to a_2. After completing the key update, the nodes in $S_{(0,0)}^{2}$ need to delete the master key and perform the following hash processing on the connection keys:

$$K_{b,c} = H_{MK_c}\left(ID_b\right) \tag{12}$$

where $b \in S_{(0,0)}^{2}$, $c \in S_{(0,0)}^{3}$.

Step 5. After receiving the information K_{a,c_3} securely, a2 replaces its vertical connection node and its vertical connection key with c3 and K_{a,c_3}, respectively.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3524409

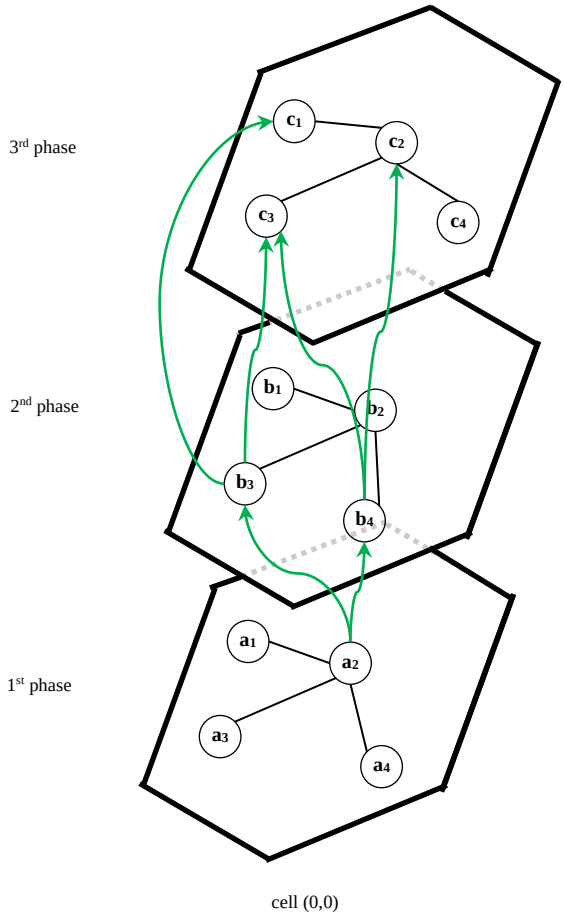


Fig. 4 Key update process.

IV. SCHEME ANALYSIS

There are two key indicators for evaluating a key management mechanism [2]-[32]: local connectivity and resilience toward node capture. The local connectivity, named PL, refers to the probability of two neighboring nodes in the network establishing a shared pairwise key [2]. Resilience toward node capture, namely Pr, by estimating the fraction of total network communications which are compromised by a capture of *x* nodes not including the communications in which the compromised nodes are directly involved [3]. In this section, we also analyzed the local connectivity and the resilience of our scheme.

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

We assume that the deployment area is flat, and that the nodes follow two-dimensional Gaussian distributions within the area. The wireless communication range of an SN is 40m. The neighbor relationship between SNs is symmetrical. That is, if *A* is a neighbor of *B*, then *B* is also a neighbor of *A*. In order to verify the performance of this scheme, in this simulation, it is assumed that an enemy randomly selects a cell as the capture cell. Throughout the entire lifecycle of the network, it is assumed that the adversary only captures nodes from the capture cell and 50% of its neighboring cells of the capture cell, and new nodes are needed to be added to the aforementioned cells only where 50% of nodes can work normally. Assuming that the lifecycle of a node is 5 phases, i.e. Gw=5. The performance of our scheme is calculated based on these nodes newly deployed in the capture cell.

A. Calculate PL^i

PL^i represents the probability that a node newly deployed at the i-th phase can establish a shared pairwise key with its neighboring nodes. P_L^i can be estimated using the following formula:

$$P_L^i = P_d^i + P_{ind}^i \tag{13}$$

where P_d^i and P_{ind}^i indicate the probabilities of pairwise key establishment between a newly deployed node and a node deployed earlier in the network in a direct or indirect manner in the ith phase, respectively. They can be estimated by the following formula:

$$\begin{aligned} P_d^i = {} & P_{SC}^i \cdot \left(P_{SC-LG}^i + \left(1 - P_{SC-LG}^i\right) \cdot P_{SC} \cdot P_{k-U}^i \right) \\ & + P_{NC}^i \cdot P_{NC-LG}^i \cdot \left(P_{ST}^i \cdot \left(1 - \left(1 - P_{NC}\right)^2\right) + \left(1 - P_{ST}^i\right) \cdot P_{NC} \right) \end{aligned} \tag{14}$$

In the above formula, P_{SC}^i and P_{NC}^i indicate the proportion of a node's neighbors whose deployment cells are the same or adjacent to that of the node deployed in the ith phase, P_{SC-LG}^i and P_{NC-LG}^i represent the proportion of newly deployed nodes in P_{SC}^i and P_{NC}^i, respectively, P_{ST}^i represents the proportion of nodes in P_{NC-LG}^i deployed simultaneously with the node, and P_{k-U}^i indicates the probability of a key being successfully updated during the key update in the ith phase.

The following conclusions can be drawn from the formula (2):

1. The value of $P_{SC}^i + P_{NC}^i$ increases with increasing *P*.

Prove: $$\iint_D f(x,y)\,dxdy = \int_0^{2\pi} \int_0^{3\sqrt{3}\cdot len/2} \frac{1}{2\pi\sigma^2} e^{-r^2/2\sigma^2} r\,dr\,d\theta$$

$$= 1 - e^{-\frac{\left(3\sqrt{3}\cdot len\right)}{2\sigma^2}} = P$$

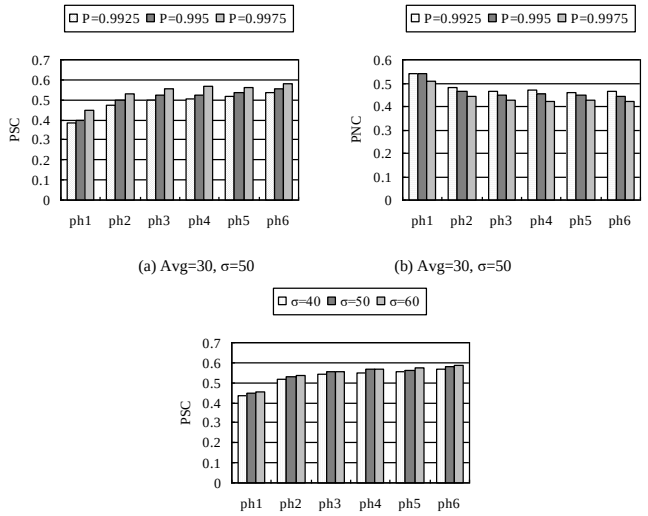
That is, the probability of a node locating in the cell and its neighbor cells of the deployment point is about *P*. Obviously, $P_{SC}^i + P_{NC}^i$ increases with increasing P. As shown in Fig.5, when $\sigma = 50$ and P =0.9925, 0.995 and 0.9975, $P_{SC}^1 + P_{NC}^1$ is about 0.93, 0.94 and 0.96 respectively.

2. When *P* is constant, P_{SC}^i increases slowly with increasing σ.

Prove: From the equation $x = \frac{3\sqrt{3}}{2} len > R$, we can find that the proportion function $f(x) = \frac{(x-R)^2}{x^2}$ is an increasing function. In addition, from the properties of the two-dimensional Gaussian distribution, it can be seen that P_{SC}^i increases with the increase of $f\left(\frac{3\sqrt{3}}{2} \cdot len\right)$. From the formula (2), it can be seen that when *P* remains constant, *Len* increases

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3524409

as σ increases. This can be confirmed from Fig. 5.



(a) Avg=30, σ=50 (b) Avg=30, σ=50

(c) P=0.9975, Avg=30

Fig. 5 P_{SC}^{i} and P_{NC}^{i} as a function parameters P, Avg and σ. In our simulations, if not specifically stated, P=0.9975, Avg=30, σ=50, P_{SC}=P_{NC}=0.4.

In our scheme, from the node capture and addition model, we can find that $P_{SC}^{i}+P_{NC}^{i}$ increases, P_{SC-LG}^{i} and P_{NC-LG}^{i} decrease after the first phase. However, after a lifecycle of nodes, the above values remain constant basically, which can be confirmed from the Fig. 6.

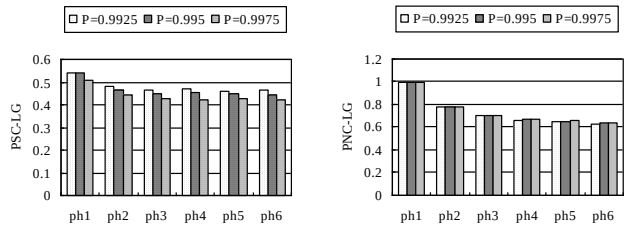


Fig. 6 P_{SC-LG}^{i} and P_{NC-LG}^{i} as a function parameter P.

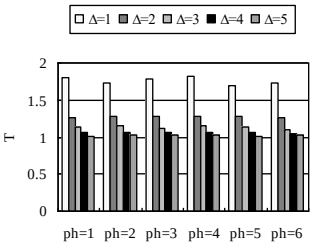


Fig. 7 T as a function parameter Δ.

From the formula (14), it can be concluded that P_{d}^{i} is related to P_{k-U}^{i}. From the formula (10), it can be concluded that the number of times required to select cluster heads decreases as Δ increases. If the cluster head selection fails, then it needs to wait for the normal clustering time before starting the next selection. So, to shorten the clustering time, Δ cannot be too small. As shown in Fig. 7, Δ should not be less than 4.

During the clustering process, the more nodes participate in broadcasting, the higher the energy consumption of the scheme. From the formula (2), it can be concluded that *Len* increases as *P* and σ increase. During the clustering process, as *Len* increases, the range of nodes that need to be broadcasted needs to be expanded, which naturally leads to an increase in the number of nodes participating in broadcasting. In addition, the increase in Avg will obviously lead to an increase in the number of nodes participating in broadcasting. All these can be confirmed from the Fig. 8.

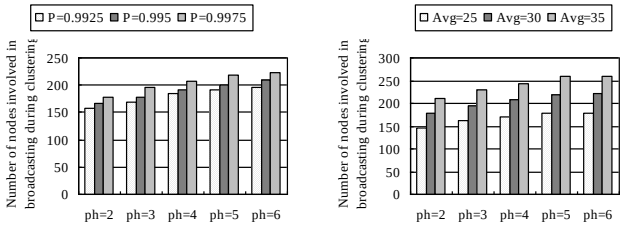


Fig. 8 The number of nodes involved in broadcasting during clusting as a function parameters P and Avg.

During the clustering process, due to differences in deployment environments, there are always very few nodes do not participate in a key update. If nodes deployed in the network do not participate in a key update, they cannot directly establish pairwise keys with nodes deployed subsequently in the network. If there *x* newly deployed nodes do not participate in a key update, the probability of keys being updated can be estimated by the following formula:

$$P_{U}=\left(1-\frac{1}{t}\right)^{x} \tag{15}$$

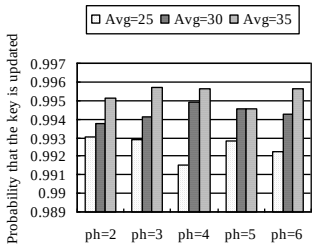


Fig. 9 Probability that a key is updated as a function parameter Avg.

From the clustering process, it can be seen that increasing Avg appropriately can help to reduce the number of nodes which need to participate in the key update. As shown in Fig. 9, when Avg increases from 25 to 30, P_{k-U}^{i} increases by about 1%. From the formula (14), it is easy to find that P_{d}^{i} increases as *P*, P_{SC} and P_{NC} increase. As shown in Fig. 10, when σ =50, $P_{SC}=0.4$, $P_{NC}=0.4$, and *P* increases from 0.9925 to 0.9975, P_{d}^{1} increases from about 0.73 to 0.77; when P=0.9975, σ =50, $P_{SC}=0.4$, and P_{NC} increases from 0.3 to 0.5, P_{d}^{1} increases form about 0.71 to 0.83; since the parameter P_{SC} does not work in the first phase, when *P*=0.9975, σ =50, $P_{NC}=0.4$, and P_{SC} increases from 0.3 to 0.5, P_{d}^{2} increases from about 0.58 to 0.61. In our scheme, nodes in the same set can directly establish

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3524409

pairwise keys. The previous analysis shows that when P_{SC} and P_{NC} do not differ much, P_d^i increases with σ increasing. As shown in Fig.10, when P=0.9975, $P_{SC}=0.4$, $P_{NC}=0.4$, and σ increases form 40 to 50, P_d^1 increases from about 0.76 to 0.77. The previous analysis shows P_d^i drops after the first phase. However, its value remains stable basically after one lifecycle of nodes, which can be confirmed from the Fig. 10.

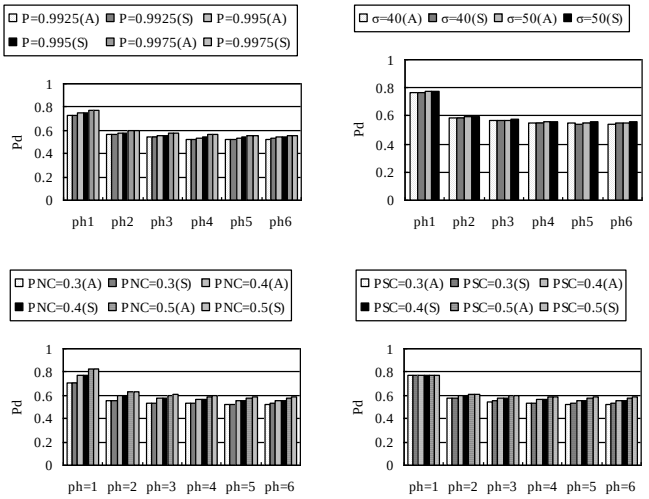


Fig. 10. P_d^i as a function parameters P_{NC}, P_{SC}, σ and P.

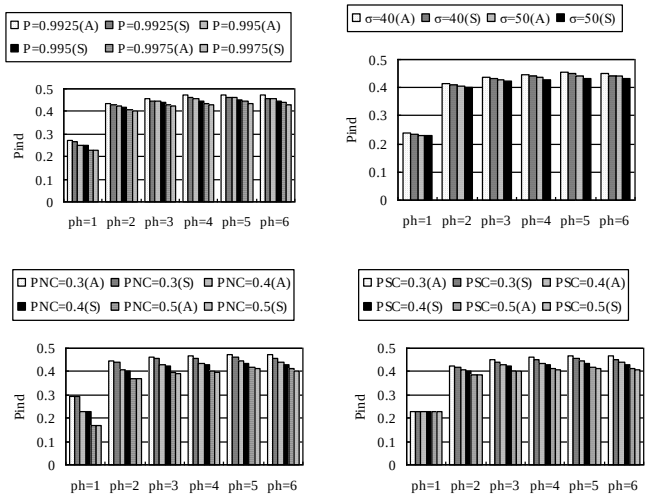


Fig. 11. P_{ind}^i as a function parameters P_{NC}, P_{SC}, σ and P.

From the process of establishing a pairwise key, it can be seen that if a newly deployed node cannot directly establish pairwise keys with its neighboring nodes, the path key establishment process needs to be initiated. The probability that a newly deployed node can establish a pairwise key with its neighboring nodes through path key establishment can be estimated by using the following formula:

$$P_{ind}^i = \left(1-P_d^i\right)\cdot\left(1-\left(1-P_d^i\right)^{Avg_{ind}^i}\right) \tag{16}$$

where Avg_{ind}^i represents the expected number of the intermediate nodes involved in the process of establishing a path key in the ith phase, which can be estimated by using the following formula:

$$\begin{aligned} Avg_{ind}^i &= \frac{\pi R^2 + 2\pi R^2 - \left(\frac{2}{3}\pi R^2 - \frac{\sqrt{3}}{2}R^2\right)}{2}\cdot\frac{Avg}{\pi R^2}\cdot\left(P_d^i\right)^2 \\ &\approx 1.03\cdot Avg\cdot\left(P_d^i\right)^2 \end{aligned} \tag{17}$$

In sensor networks, to ensure network connectivity, Avg cannot be set too small. From the formulas (16) and (17), it can be concluded that newly deployed nodes can establish path keys with a high probability for their neighboring nodes cannot directly establish pairwise keys by setting appropriate parameters. This can be confirmed from the Fig.11.

In summary, if the values of parameters are set appropriately, $P_L^i \approx 1$. This can be confirmed from Fig.10 and Fig.11.

B. Calculate Pr^i

Pr^i represents the value of Pr in the ith phase, which can be calculated using the following formula:

$$\mathrm{Pr}^i = \mathrm{Pr}_d^i + \mathrm{Pr}_{ind}^i \tag{18}$$

where Pr_d^i and Pr_{ind}^i represent the probabilities of a direct key and a path key being compromised respectively. In our scheme, the pairwise key established between nodes within the same set cannot be compromised; similarly, the pairwise key directly established between two nodes within two adjacent cells cannot be compromised too. That is, the pairwise key established between two neighboring nodes can be compromised only when the pairwise key is established by using the updated key. Therefore, Pr_d^i can be estimated using the following formula:

$$\mathrm{Pr}_d^i = \begin{cases} 0, & i=1 \\ \dfrac{P_{d\_pre}^{i'-i}\cdot\mathrm{Pr}_{d\_pre}^{i'-i}}{P_L^i}, & i>1, 0<i'<i \end{cases} \tag{19}$$

where $P_{d\_pre}^{i'-i}$ represents the probability of a pairwise key being established between two neighboring nodes by using the updated keys. $\mathrm{Pr}_{d-pre}^{i'-i}$ represents the probability of an updated key being compromised, which can be estimated by using the following formula:

$$\mathrm{Pr}_{d-pre}^{i'-i} = 1-\left(1-\frac{1}{t}\right)^{CC\times(i-i')} \tag{20}$$

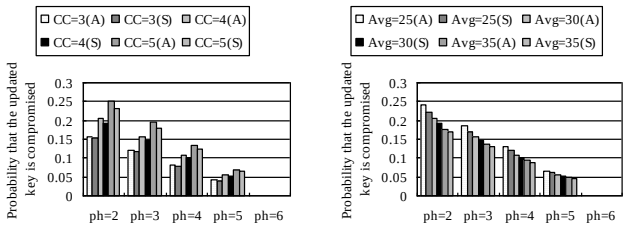


Fig. 12 Probability that a updated key is compromised as a function parameters Avg and CC.

The formula (20) shows that $\mathrm{Pr}_{d-pre}^{i'-i}$ decreases as CC decreases. As shown in Fig.12, when P=0.9975, $\sigma = 50$ and

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3524409

Avg=30, *CC* decreases from 5 to 3, $\Pr_{d-pre}^{2-6}$ drops from about 0.24 to 0.15. For sensor networks, the value of CC is difficult to set. To this end, appropriately increasing t is a feasible approach. The previous analysis shows that t increases as P, σ and Avg increase. As shown in Fig.12, when P=0.9975, $\sigma = 50$ and Avg increases from 25 to 35, $\Pr_{d-pre}^{2-6}$ drops from about 0.23 to 0.17.

$\Pr_{d-pre}^{i'-i}$ increases as $i-i'$ increases. Similarly, the node capture model and deployment model show that $\frac{P_{d\_pre}^{i'-i}}{P_L^i}$ decreases as $i-i'$ increases. The formulas (19) and (20) show that $\Pr_d^i$ can be very small by setting appropriate parameters. As shown in the Fig.13, $\Pr_d^i$ is never greater than 0.01. The formula (20) shows that $\Pr_{d-pre}^{i'-i}$ is independent of P_{SC}. However, the previous analysis shows that P_d^i increases as P_{SC} increases. From formula (19), it is concluded that $\Pr_d^i$ decreases as P_{SC} increases. As shown in Fig.13, when P=0.9975, $\sigma = 50$, Avg=30, P_{NC}=0.4, P_{SC} increases from 0.3 to 0.5, $\Pr_d^6$ increases from about 0.005 to 0.009. However, after a lifecycle of nodes, parameters in (19) and (20) remain stable basically, $\Pr_d^i$ also remains stable basically. This can be confirmed from the Fig.13.

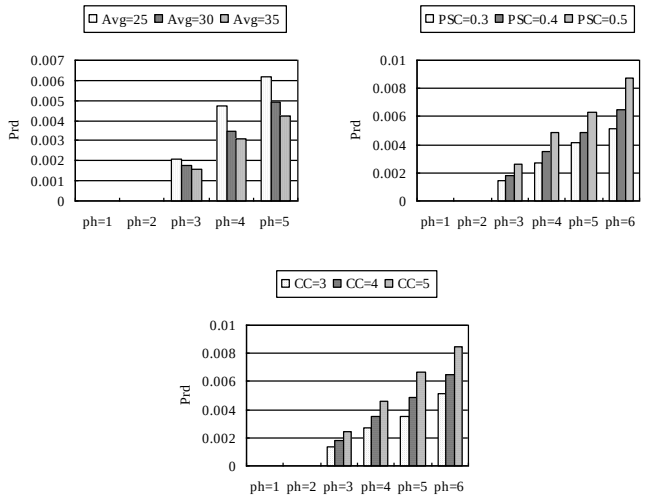


Fig. 13. $\Pr_d^i$ as a function parameters P_{SC}, Avg and CC.

$\Pr_{ind}^i$ can be estimated using the following formula:

$$\Pr_{ind}^i = \frac{P_{ind}^i}{P_L^i} \cdot \left(P_C\right)^{unAvg_{ind}^i} \tag{21}$$

where P_C is the probability of a node's neighbors being compromised, $unAvg_{ind}^i$ is the expected number of trusted intermediate nodes involved in the path keys establishment of the ith phase and it can be estimated using the following formula:

$$unAvg_{ind}^i = Avg_{ind}^i \times \left(1-\Pr_d^i\right) \tag{22}$$

The formula (21) shows that when P_C is fixed, $\Pr_{ind}^i$ declines exponentially with increasing $unAvg_{ind}^i$. From the formulas (18) to (22), it is easy to find that P_{SC} and *Avg* have a greater impact on $\Pr_{ind}^i$ as compared with *CC*. As shown in Fig.14, when P=0.9975, $\sigma = 50$, P_{NC}=P_{SC}=0.4 and Avg increases from 25 to 35, $\Pr_{ind}^i$ declines from about 0.022 to 0.012. When P=0.9975, $\sigma = 50$, Avg=30, P_{NC}=0.4 and P_{SC} increases from 0.3 to 0.5, $\Pr_{ind}^i$ declines from about 0.023 to 0.011. However, when P=0.9975, $\sigma = 50$, Avg=30, P_{NC}=P_{SC}=0.4, and CC increases from 3 to 5, $\Pr_{ind}^6$ increases from about 0.015 to 0.017.

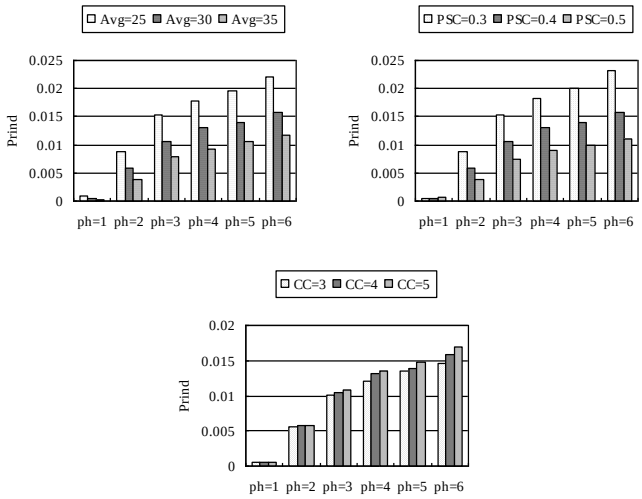


Fig. 14. $\Pr_{ind}^i$ as a function parameters P_{SC}, Avg and CC.

In order to comprehensively evaluate this scheme, we introduced the parameter of resilient local connectivity [34]. Resilient local connectivity is the probability that two neighboring nodes can establish a secure pairwise key between them under capture attacks. This metric naturally considers both local connectivity and resilience. This is a very important indicator because after capture attacks, the number of nodes that can function normally in the network will decrease. If secure communication keys cannot be established with high probability between normal nodes, it may lead to the entire network being divided into multiple secure connected sets. Although any two nodes in a secure connectivity set have a secure path, there is no secure path between nodes in different secure connectivity sets. The research on network security connectivity has exceeded the scope of this article, please refer to [35] for details. Resilient local connectivity can be estimated using the following formula:

$$\Pr_L^i = P_d^i \times \left(1-\Pr_d^i\right) + P_{ind}^i \times \left(1-\Pr_{ind}^i\right) \tag{23}$$

From the previous analysis and Fig. 10, Fig. 11, Fig. 13, and Fig. 14, it can be seen that when P=0.9975, σ=50, Avg=30, P_{NC}=P_{SC}=0.4, and CC=5, $\Pr_L^6$ is about 0.977. From the conclusion in [35], it can be concluded that in the above situation, the probability of normal nodes in the network being safely connected is about 1.

C. Comparison With the State-of-the-Art Technique

In this section, local connectivity and resilience against node

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3524409

capture attacks of our scheme, POK scheme [17], OWAKM scheme [20] and RGM scheme [27] are compared. In our simulation, for fairness, the POK, OWAKM and RGM schemes also uses the same node deployment, capture, and addition model as our scheme, and the number of nodes compromised during the key establishment initialization phase is the same.

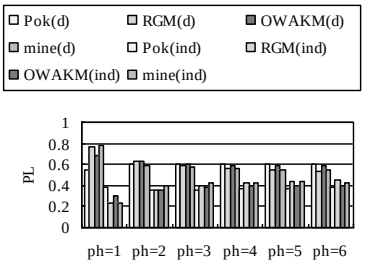


Fig. 15 Comparisons of the value of P_L^i of the POK, OWAKM, RGM and our scheme.

Fig. 15 shows a comparison of local connectivity among three schemes. At the first phase, it can be concluded from Fig. 15 that the POK scheme has the smallest PL_d^1, and our scheme is almost identical to the RGM scheme, while the OWAKM scheme is located in the middle. In the POK scheme, nodes whose deployment points are adjacent can establish pairwise keys only when they are pre-distributed the same keys. That is, when $P_{NC} = 0.4$, the probability that two nodes from two neighboring cells, which can establish a pairwise key directly is $\left(P_{NC}\right)^2 = 0.16$. In our scheme, the probability that two nodes deployed in two adjacent cells can establish a pairwise key is $1\text{-}\left(1\text{-}P_{NC}\right)^2 = 0.64$. Therefore, with the same parameter settings, in the first deployment phase, the local connectivity of our scheme is higher than that of the POK scheme. In addition, by setting appropriate parameters, PL_d^1 of the RGM scheme is almost identical to our scheme. After the first phase, due to the key pool of the POK scheme are constructed by using backward key chains, nodes deployed later can directly establish pairwise keys with nodes deployed earlier as long as they are pre-ditributed the same key chains. The previous analysis shows that $PL_d^2 > PL_d^1$, but after the second phase, the value of PL_d^i remains stable basically. This can be confirmed from Fig. 15. The key pool of OWAKM scheme is constructed using three-dimension backward keychains. Due to the fact that the third key of a three-dimensional reverse keychain is different at each deployment phase. That is to say, only nodes deployed at the same phase can use the third-dimension keys of the keychains to establish a shared key. This leads to $PL_d^2 < PL_d^1$. But when the members of neighboring nodes remain stable, the PL_d^i of this scheme remains basically stable. This can be confirmed from Fig. 15. In RGM scheme, due to independent key pool is used at each phase, nodes deployed at the same phase can use pre-distribution keys to establish pair-wise keys, nodes deployed at different phases only can use partially pre-distribution keys to establish pair-wise keys. This leads to a decrease in PL_d^i as i increases. This can also be confirmed from Fig. 15. Furthermore, from Fig. 15, it can also be concluded that after one lifecycle, its local connection remains stable and nodes can establish pairwise keys with neighboring nodes with high probability. As shown in Fig. 15, $PL^i = PL_d^i + PL_{ind}^i$ of all three above schemes are greater than 0.98.

In the POK scheme, the multi-phase key pools are constructed by using backward key chains. From the characteristics of backward key chain technology, it can be seen that if a key of a key chain of the ith phase is compromised, the keys before the i-th phase of the keychain are no longer secure. Like the POK scheme, the OWAKM scheme also uses backward keychains to build key pools, so the resilience of this scheme also decreases with the increase of deployment phases. In RGM, each deployment phase uses an independent key pool, and the nodes deployed in the i-th phase need to store keys from the key pools of the i-th phase to the (i+Gw-1)th phase, and the keys selected from the key pools of the (i+1)th phase to the (i+Gw-1)th phase are hashed before stored in the nodes. Therefore, nodes compromised in the i-th phase will only pose a threat to the secure communication of nodes deployed in the same phase. In our scheme, the use of the unique pairwise key for communication between two nodes further enhances the resilience of the RGM scheme. When more nodes are compromised during the key establishment, the resilience of our scheme is more prominent. When CC=8, $\Pr^6$ of POK scheme, RGM scheme, OWAKM scheme, and our scheme are approximately 0.21, 0.15, 0.29 and 0.05, respectively.

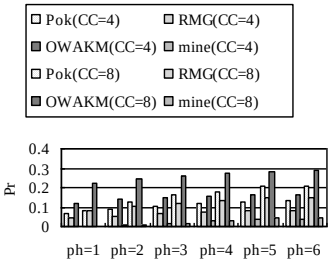


Fig. 16 Comparisons of $\Pr^i$ of POK, OWAKM, RGM and our scheme.

Based on the previous analysis, when CC=8, $\Pr_L^6$ of POK scheme, RGM scheme, OWAKM scheme, and our scheme are approximately 0.78, 0.84, 0.7 and 0.94, respectively.

V. CONCLUSIONS

In this paper, for multi-phase deployment sensor networks, we propose a new pairwise key scheme based on deployment knowledge. In this scheme, the key pool of each cell and the key pool of each deployment phase are independent of each other. These nodes with the same deployment point, deployed in different phases, can communicate safely by updating keys on-line. In addition, we proposed a new path key establishment method, which can further improve the probability of establishing a pairwise key between nodes. Theoretical analysis and simulation show that our scheme can not only establish a pairwise key between nodes with high probability, but also enables the network to have a good resilience against node capture attacks by setting appropriate parameters. For example,

in our simulation, even in the 6th phase, the probability of establishing a pairwise key between normal working nodes is not less than 0.98, and the probability of pairwise keys being compromised is not exceed 0.05.

VI. 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] D. Liu, P. Ning and R. Li, "Establishing Pairwise Keys in Distributed Sensor Networks," J. ACM Computer and Comm. Security, Vol. 8, No. 1, pp 41–77, 2005.

[6] 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.

[7] 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.

[8] F. Gandino, R. Ferrero, and M. Rebaudengo, "A key distribution scheme for mobile wireless sensor networks: q-s-composite," IEEE Transactions on Information Forensics and Security, vol. 12, no. 1, pp. 34–47, 2017.

[9] 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.

[10] U. Altun, S. T. Basaran, G. K. Kurt, et al., "Scalable Secret Key Generation for Wireless Sensor Networks," IEEE SYSTEMS JOURNAL, vol. 16, no. 4, pp. 6031-6041, 2022.

[11] A. Singh1, K. Jain, "An efficient secure key establishment method in cluster-based sensor network," Telecommunication Systems , vol. 79, pp. 3–16, 2022.

[12] A. Shukla, S. Tripathi, M. S, et al., "SEE2PK: Secure and energy efficient protocol based on pairwise key for hierarchical wireless sensor network," Peer-to-Peer Networking and Applications, vol. 17, pp. 701–721, 2024.

[13] S. A. Khah, A. Barati, H. Barati, "A dynamic and multi-level key management method in wireless sensor networks (WSNs)," Computer Networks, vol. 236, 109997, 2023.

[14] 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.

[15] 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.

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

[17] M. L. Messai, "A Self-Healing Pairwise Key Pre-Distribution Scheme in IoT-based WSNs," In: IEEE International Wireless Communications and Mobile Computing (IWCMC), pp. 904-909, 2023.

[18] 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.

[19] 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.

[20] 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.

[21] C. Castelluccia, A. Spognardi, "Rok: A robust key pre-distribution protocol for multi-phase wireless sensor networks", In 2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007, pp. 351-360.

[22] H. Ito, A. Miyaji, K. Omote, "RPoK: A strongly resilient polynomialbased random key pre-distribution scheme for multiphase wireless sensor networks", In 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, pp. 1-5.

[23] S. Sarimurat, A. Levi, "Hag: Hash graph based key predistribution scheme for multiphase wireless sensor networks", In 2013 IEEE International Conference on Communications (ICC), pp. 2079-2083.

[24] M. L. Messai, Hamida Seba, "A Self-healing Key Pre-Distribution Scheme for Multi-Phase Wireless Sensor Networks," IEEE Trustcom/BigDataSE/ICESS, pp. 144-151, 2017.

[25] A. Durresi, V. Bulusu, V. Paruchuri, "SCON: Secure management of continuity in sensor networks," Computer Communications, Vol. 29, no. 13–14, pp. 2458-2468, 2006.

[26] 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.

[27] M. Ergun, A. Levi, E. Savas, "Increasing resiliency in multi-phase wireless sensor networks: generationwise key predistribution approach," The Computer Journal, vol. 54, no. 4, 602–616, 2011.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

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

[34] 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, pp. 1370 – 1381, 2011.

[35] Zhao J; On Secure Communication in Sensor Networks Under q-Composite Key Predistribution With Unreliable Links, IEEE Transactions on Communications, vol. 70, no. 2, 2022.

Sujun Li received her M. S. degree in computer science from Hunan University, China, in 2008. Her main research interest is sensor networks and information security.

Boqing Zhou received his Ph. D. degree in computer science from Hunan University, China. Currently, he is a postdoctoral student with the School of Information Science and Engineering, Central South University, Changsha Hunan PR China. His current main research interests include sensor networks and information security.

Decheng Miao received his PhD degree in computer application technology from South China University of Technology, China, in 2012. Currently, he is working as a professor at the School of Information Engineering, Shaoguan University, Shaoguan, Guangdong, PR China. His current research interests include formal methods, categorical theory, database and networking. He has published more than 50 papers in various International journals and refereed conferences.

Yun Cheng received his PhD degree in computer science from National University of Defense Technology, China, in 2006. Currently, he is working as a professor at the school of information, Hunan University of Humanities, Science and Technology, Loudi, Hunan, PR China. His current research interests include algorithm analysis and optimization, computer network and optical communication technology.

Jie Wu is the Director of the Center for Networked Computing and Laura H. Carnell professor at Temple University. He also serves as the Director of International Affairs at the College of Science and Technology. He served as Chair of the Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Asscoaite Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, network trust and security, distributed algorithms, applied machine learning, and cloud computing. Dr. Wu regularly published in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Mobile Computing, IEEE Transactions on Service Computing, Journal of Parallel and Distributed Computing, and Journal of Computer Science and Technology. Dr. Wu is/was general chair/co-chair for IEEE DCOSS'09, IEEE ICDCS'13, ICPP'16, IEEE CNS'16, WiOpt'21, ICDCN'22, IEEE IPDPS'23, and ACM MobiHoc 2023 as well as program chair/cochair for IEEE MASS'04, IEEE INFO COM'11, CCF CNCC'13, and ICCCN'20. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a Fellow of the AAAS and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.