Hindawi Publishing Corporation
Journal of Sensors
Volume 2014, Article ID 573913, 10 pages
http://dx.doi.org/10.1155/2014/573913

Research Article

A New Key Predistribution Scheme for Multiphase Sensor Networks Using a New Deployment Model

Boqing Zhou,[1,2] Jianxin Wang,[2] Sujun Li,[1,2] and Weiping Wang[2]

[1] *Hunan University of Humanities, Science and Technology, Loudi, Hunan 417000, China*
[2] *School of Information Science and Engineering, Central South University, Changsha, Hunan 410083, China*

Correspondence should be addressed to Jianxin Wang; jxwang@mail.csu.edu.cn

Received 1 May 2014; Accepted 18 May 2014; Published 15 June 2014

Academic Editor: Athanasios V. Vasilakos

During the lifecycle of sensor networks, making use of the existing key predistribution schemes using deployment knowledge for pairwise key establishment and authentication between nodes, a new challenge is elevated. Either the resilience against node capture attacks or the global connectivity will significantly decrease with time. In this paper, a new deployment model is developed for multiphase deployment sensor networks, and then a new key management scheme is further proposed. Compared with the existing schemes using deployment knowledge, our scheme has better performance in global connectivity, resilience against node capture attacks throughout their lifecycle.

1. Introduction

Due to limited energy capacity of batteries and the possibility of node capture, the functional lifetime of sensor networks (SNs) generally is longer than the operational lifetime of single node. To keep networks working efficiently, multiple deployments of nodes are needed. In the paper, multiphase SNs (MSNs) are studied, in which new nodes are periodically redeployed with certain intervals, called multiphase, to replace the dead or compromised nodes.

When SNs are deployed in a hostile environment, security becomes extremely important as they are vulnerable to different types of malicious attacks [1–4]. 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 using cryptographic techniques.

Public-key operations (both software and hardware implementations), albeit computationally feasible [5, 6], consume energy approximately *three orders of magnitude* higher than symmetric key encryption [7]. Therefore, in the last few years, different key distribution schemes using symmetric key algorithms have been developed for SNs [8–26].

However, the security issue is still not solved for MSNs by using deployment knowledge. In the schemes [16, 17], a fraction of keys known by an attacker increases with the capture of nodes due to the repeated use of a fixed key pool. As a result, network security significantly declines with time. When a certain number of these nodes are captured, the adversary has enough keys to compromise a large number of links making the network ineffective. Addition of new nodes to the network with keys from the same key pool will not help because the keys in the new nodes are compromised. In [20], a multiphase key management scheme is proposed, in which a multiphase deployment model is used. However, it has the following shortcomings. (1) In a cell, only a few nodes which are not captured are working in a long time. (2) Nodes must know their location information. (3) The number of new nodes added to the network is fixed in every deployment, which will give rise to the number of nodes uncaptured in the network with time. Also, the key management scheme proposed based on the deployment model has the following shortcomings. (1) Nodes which reside in the same cell but are deployed in different phases cannot communicate with each other. As a result, the local connectivity is low. (2) The *global connectivity* will significantly decrease with time.

1.1. Outline of Our Scheme. To sum up, the problem of authentication and pairwise key establishment between nodes is still not solved for MSNs. In this paper, the main focus is twofold. (1) A new multiphase deployment model is proposed for sensor networks. In the model, the deployment field is divided into hexagonal cells, each cell has a deployment point, and nodes which have the same point form a group. When the proportion of uncaptured nodes in a group is less than the threshold ρ_0, new nodes are needed to be added to the cell. (2) A new key management scheme is proposed based on the deployment model. In our scheme, network deployment includes n phases. For a cell, a disjoint and association n phases' key pool is created, which is generated by two-dimension backward key chains [21]. Key pool of each phase is divided into 7 equal size subkey pools. And nodes deployed in the ith phase and deployed in a cell (r, c) pick keys from the ith-phase key pool of the cell (r, c) and key pools which are created by neighbors cells of the cell (r, c).

1.2. Main Contributions. The main contributions of this paper are summarized as follows.

(1) A multiphase deployment model is presented. The model has the following two main advantages: (1) the number of nodes which are not captured in a cell can be controlled by adjusting the parameter ρ_0; (2) nodes do not need to know their location information.

(2) A new method to construct key pools is proposed and a new key predistribution scheme is presented. The scheme can provide good performances in local connectivity, global connectivity, and resilience against node capture.

1.3. Organization. The remainder of the paper is organized as follows. The existing schemes are summarized in Section 2. The model of deployment is introduced in Section 3. Our approach is proposed in Section 4 and the analysis and simulation results are provided in Section 5. Conclusion and future work are given in Section 6.

2. Related Work

To improve the performance of key establishment, Du et al. [16] and Yu and Guan (YG scheme) [17] developed a scheme using predeployment knowledge, respectively. In [16], the network area is divided into a grid and information on the associated matrices is stored in the sensors based on deployment knowledge. In [17], the network area is divided into hexagonal cells. Compared with [16], the scheme achieves a higher connectivity with a much lower memory requirement and a shorter transmission range. In the two schemes, all nodes choose their keys from the same key pool. An attacker can easily obtain a large number of keys by capturing a small fraction of nodes, which can make SNs ineffective. The addition of new nodes to the network with keys from the same key pool will not help because the keys in the new nodes are already compromised. Therefore, for MSNs, the above two schemes are ineffective.

For MSNs, in [20], a scheme (ESPK scheme) is proposed using deployment knowledge, in which a multiphase deployment model is presented. In the model, the deployment field is divided into a grid. Each cell has a deployment point. Nodes which have the same deployment point form a group. The number of nodes in a group is N. And it is supposed that a new group of nodes are needed to be added to a cell only when 90% of nodes in the cell are captured. The model has the following shortcomings. (1) To know the number of nodes in each cell, location information of nodes is needed. (2) If just 80% of nodes in a cell are captured, there are a few nodes in the cell that are working in a long time. (3) The number of new nodes added to the network is measured in a group, and the number of nodes in a group is fixed, which will give rise to the number of nodes in the network with time. On the other hand, the proposed key management scheme can provide good resilience against node capture by using disjoint key pools. However, nodes which come from different phases but are deployed in the same cell cannot establish shared keys. As a result, the local connectivity is low, and the global connectivity decreases significantly with time. So, the problem of secure is still not solved for MSNs using deployment knowledge.

3. Deployment Knowledge and Threat Models

3.1. Multiphase Deployment Knowledge Model. As shown in Figure 1, a target field is partitioned into hexagon cells, and each cell has a deployment point that resides in the center of the cell. Node distribution follows two-dimensional Gaussian distributions [27] with the deployment point as center.

Nodes which are deployed in the same cell form a group. And nodes deployed in the cell (r, c) are denoted by $G_{(r,c)}$. The number of nodes in $G_{(r,c)}$ is N. $G_{(r,c)}$ is clustered into phases according to the deployment time. The i-phase subgroup of $G_{(r,c)}$ is denoted by $G^i_{(r,c)}$. In our scheme, $\mathrm{SN}_{(r,c)}$ represents the set of nodes whose deployment point locate in the cell (r, c) and that are not captured, and $|\mathrm{SN}_{(r,c)}| \le N$ (several schemes have been proposed to identify the compromised sensors in prior studies, such as [28]). When $\rho_{(r,c)}$ is less than the threshold ρ_0, we should add N-$\mathrm{SN}_{(r,c)}$ new nodes to the cell. The $\rho_{(r,c)}$ can be calculated as follows:

$$\rho_{(r,c)} = \frac{|\mathrm{SN}_{(r,c)}|}{N}. \quad (1)$$

In a deployment phase, if no new nodes are needed to be added to a cell, then the number of deployment phase of the cell remains unchanged. For example, in the second deployment phase, no new nodes are needed to be added to the cell (1, 1); the number of recent deployment phase of the cell is 1 not 2.

3.2. Threat Model. Due to the short time period of the direct key establishment phase, it is reasonable to believe that only a limited number of sensor nodes may be compromised by an attacker [2, 20–23]. We further assume that if an attacker

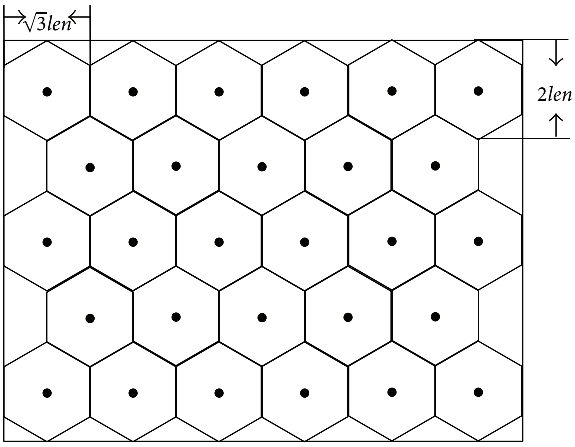


Figure 1: A target field is partitioned into hexagon grids. ● represents a deployment point.

captures a node, all the keying information it holds will also be compromised.

In the scheme, the attack model is similar with [16], when an attacker locates in a cell, he can capture nodes around it.

4. Our Scheme

4.1. Two-Dimensional Backward Key Chain. In [21], a two-dimensional backward key chain is constructed (see Figure 2). For a two-dimensional backward key chain C_j, if the key $k_j^{i_1}$ is known, the key $k_j^{i_2}$ ($i_2 \le i_1$), the generation key $g_j^{i_2}$, and the first key $k_j^{(i_2,0)}$ of the second dimensional key chain can be calculated as follows, respectively: $k_j^{i_2} = H_1^{i_1-i_2}(k_j^{i_1})$, $g_j^{i_2} = H_2(k_j^{i_2},0)$, and $k_j^{(i_2,0)} = H_2(k_j^{i_2},1)$, where H_1 and H_2 are two independent hash functions. So, the key $k_j^{(i_2,l_2)}$ ($l_2 \ge 1$) can be computed as follows:

$$k_j^{(i_2,l_2)} = H_2^{l_2}\left(g_j^{i_2}, k_j^{(i_2,0)}\right), \quad \text{when } l_2 \ge 1. \tag{2}$$

If the keys $k_j^{i_1}$ and $k_j^{(i_2,l_1)}$ are known, the key $k_j^{(i_2,l_2)}$ ($l_1 < l_2$) can be computed using the following equation:

$$k_j^{(i_2,l_2)} = H_2^{l_2-l_1}\left(g_j^{i_2}, k_j^{(i_2,l_1)}\right), \quad \text{when } l_2 > l_1. \tag{3}$$

4.2. Key Pool. In our scheme, the key pool is made up by two-dimensional backward key chains [21]. The key pool of the cell (r,c), namely, $P_{(r,c)}$, which consists of m two-dimensional backward hash key chains, is divided into$n$ phases according to the generation of the keys. $P_{(r,c)}^i$ represents the ith-phase key pool of the cell (r,c). $P_{(r,c)}^i$ is divided into seven equal size subkey pools, and $P_{(r,c)_s}^i$ represents the sth ($0 \le s \le 6$) pool (see Figure 3). $P_{(r,c)_s}^i$ consists of the following two parts: one is a generation key pool $Pg_{(r,c)_s}^i = \{k_j^i, j \in [1,m]\}$ and the other is an ordinary key pool $Pc_{(r,c)_s}^i = \{k_j^{(i,l)},\ j \in [1,m],\ 1 \le l \le L\}$.

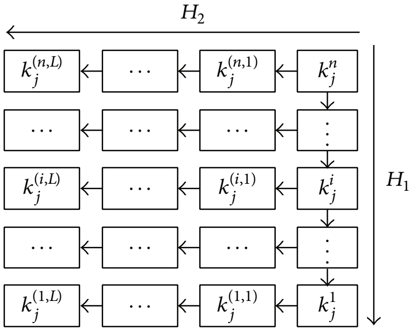


Figure 2: Two-dimensional key chain.

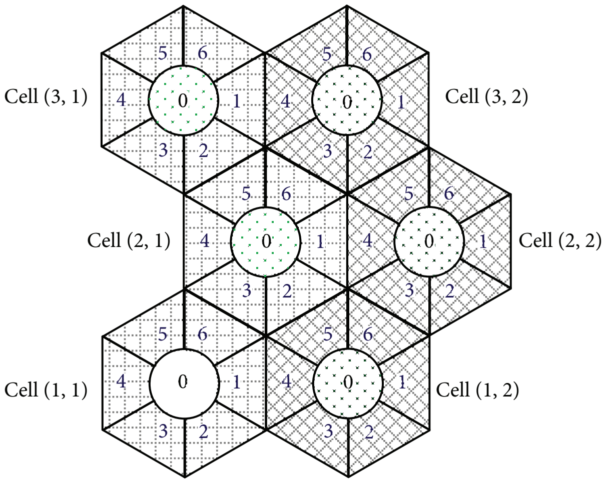


Figure 3: Subkey pools.

4.3. Our Scheme. Our scheme consists of three phases: key predistribution phase, shared-key discovery phase, and path-key establishment phase. Although path-key establishment phase is the same as, key predistribution phase and share-key discover phase are different in the previous schemes [16, 17, 20]. The details of our scheme are described below.

4.3.1. Key Predistribution Phase. This phase is conducted offline before sensor nodes are deployed. A node $a_{(r,c)}^i$ deployed in the ith deployment phase and its deployment point locates in the cell (r,c), is predistributed the following keys.

Step 1. Select randomly and uniformly $t1$ ($t1$ is a system parameter) keys from $Pc_{(r,c)_s}^i (1 \le s \le 6)$. In this step, the total number of predistribution keys of the node is $6 \times t1$.

Step 2. Select randomly and uniformly $t2$ ($t2$ is a system parameter) keys from $Pc_{(r,c)_0}^i$.

Step 3. Select randomly and uniformly $t3$ ($t3$ is a system parameter) keys from $Pg_{(r,c)_0}^i$, and meet the following condition: the number of keys from a two-dimensional backward hash key chain is no more than 1. For example, it is supposed that $k_j^{(i,l_1)}$ has been predistributed to $a^i_{(r,c)}$ and k_j^i cannot be predistributed to $a^i_{(r,c)}$.

Step 4. Select randomly and uniformly $t3$ keys from $Pg^{i'}_{(r',c')_{s'}}$, where (r', c') is the neighbor cell of (r, c), $Pg^{i'}_{(r',c')_{s'}}$ represents the generation key pool of the cell (r', c'), and i' denotes the recent deployment phase. For example, if the deployment phase of cells $(1, 1)$ and $(1, 2)$ is 2 and 3, respectively, and new nodes are needed to be added to cell $(1, 2)$, but no new nodes are needed to be added to cell $(1, 1)$; then $a^4_{(1,2)}$ should pick $t3$ keys from $Pg^2_{(1,1)_1}$ (see Figure 3). But if both cells $(1, 1)$ and $(1, 2)$ need to add new nodes, then $a^4_{(1,2)}$ and $b^3_{(1,1)}$ should pick $t3$ keys from $Pg^3_{(1,1)_1}$ and $Pg^4_{(1,2)_4}$ (see Figure 3), respectively.

4.3.2. Shared-Key Discovery Phase. In our scheme, after shared key establishment, each node should save the hashed keys in its key ring. For example, it is supposed that an sensor $a^i_{(r,c)}$ is pre-distributed two keys $k^i_{j_1}$ and $k^{(i,l)}_{j_2}$. As soon as the shared keys establishment between $a^i_{(r,c)}$ and other nodes is finished, $a^i_{(r,c)}$ saves the two following hashed keys: $H_2(k^i_{j_1}, \mathrm{ID}_{a^i_{(r,c)}})$ and $H_2(k^{(i,l)}_{j_2}, \mathrm{ID}_{a^i_{(r,c)}})$, where $\mathrm{ID}_{a^i_{(r,c)}}$ is the identity of node $a^i_{(r,c)}$.

Next, we will describe the method for any two nodes $a^{i_1}_{(r_1,c_1)}$ and $b^{i_2}_{(r_2,c_2)}$ (in the following analysis, a and b are short for $a^{i_1}_{(r_1,c_1)}$ and $b^{i_2}_{(r_2,c_2)}$) to establish a shared key at length. Similarly, in the following analysis, it is supposed that $T_a \geq T_b$, where T_a and T_b represent the deployment time of nodes a and b, respectively. And in our scheme, if nodes a and b are deployed in the same cell or the neighboring cells, then they can establish a pairwise key, otherwise, they cannot.

If $r_1 = r_2$ and $c_1 = c_2$, when $T_a = T_b$, the values of i_1 and i_2 are equal. The pairwise key between them consists of the following three parts (see Figure 4(a)): (1) $x1$ generation keys, $k^{i_1}_{j_1}, \ldots, k^{i_1}_{j_{x1}}$, which come from the generation key pools; (2) $x2$ ordinary keys, $k^{(i_1,l_1)}_{j'_1}, \ldots, k^{(i_1,l_{x2})}_{j'_{x2}}$, which come from the ordinary key pools $Pc^{i_1}_{(r_1,c_1)_s}$ $(0 \leq s \leq 6)$; (3) $x3$ ordinary keys, which come from the ordinary key pool $Pc^{i_1}_{(r_1,c_1)_0}$. For example, let keys $k^{i_1}_{j''_1}, \ldots, k^{i_1}_{j''_{l_{x'}}}, k^{(i_2,l'_1)}_{j''_{l_{x'}+1}}, \ldots, k^{(i_2,l'_{x3-l_{x'}})}_{j''_{x3}}$ and $k^{(i_1,l_1)}_{j''_1}, \ldots, k^{(i_1,l_{x'})}_{j''_{l_{x'}}}, k^{i_2}_{j''_{x'+1}}, \ldots, k^{i_2}_{j''_{x3}}$ be predistributed to nodes a and b, respectively. Nodes a and b can calculate the keys $k^{(i_1,l_1)}_{j''_2}, \ldots, k^{(i_1,l_{x'})}_{j''_{l_{x'}}}$ and $k^{(i_2,l'_1)}_{j''_{l_{x'}+1}}, \ldots, k^{(i_2,l'_{x3-l_{x'}})}_{j''_{x3}}$ by using the keys $k^{i_1}_{j''_1}, \ldots, k^{i_1}_{j''_{l_{x'}}}$ and $k^{i_2}_{j''_{x'+1}}, \ldots, k^{i_2}_{j''_{x3}}$ and the method described in Section 4.1. When $T_a \neq T_b$ (without loss of generality, it is supposed that $T_b < T_a$), the pairwise key between them consists of the following two parts (see Figure 4(b)). (1) $x1$ hashed generation keys, $H(k^{i_1}_{j_1}, \mathrm{ID}_b), \ldots, H(k^{i_1}_{j_{x1}}, \mathrm{ID}_b)$. Node a can calculate these keys by using the predistributed keys $k^{i_2}_{j_1}, \ldots, k^{i_2}_{j_{x1}}$ and the method described in Section 4.1 and in Section 4.3.2. (2) $x3$ hashed ordinary keys, $H(k^{(i_1,l_1)}_{j''_1}, \mathrm{ID}_b), \ldots, H(k^{(i_1,l_{x'})}_{j_{l_{x3}}}, \mathrm{ID}_b)$. Node a can calculate these keys by using the predistributed keys $k^{i_2}_{j''_1}, \ldots, k^{i_2}_{j''_{x1}}$ from the key pool $Pg^{i_1}_{(r_1,c_1)_0}$ and the same method as the previous.

If cells (r_1, c_2) and (r_2, c_2) are neighbor cells, the values of $x1$ and $x2$ are equal to 0. When $T_a = T_b$, shared keys between them include x' and $x'' = x3 - x'$ ordinary keys (see Figure 5(a)), $k^{(i_1,l_1)}_{j_1}, \ldots, k^{(i_1,l_{x'})}_{j_{x'}}$ and $k^{(i_2,l'_1)}_{j'_1}, \ldots, k^{(i_2,l'_{x3-x'})}_{j'_{x3-x'}}$, which come from the ordinary key pool $Pc^{i_1}_{(r_1,c_1)_{sa}}$ $(sa \in [1, 6])$ and $Pc^{i_2}_{(r_2,c_2)_{sb}}$ $(sb \in [1, 6])$, respectively. Nodes b and a can calculate these keys by using these predistributed keys $k^{i_1}_{j_1}, \ldots, k^{i_1}_{j_{x'}}$ and $k^{i_2}_{j'_1}, \ldots, k^{i_2}_{j'_{x3-x'}}$, respectively. When $T_a \neq T_b$ (without loss of generality, it is supposed that $T_b < T_a$), common keys between them contain $x3 = x'$ hashed ordinary keys (see Figure 5(b)), $H(k^{(i_1,l_1)}_{j_1}, \mathrm{ID}_b), \ldots, H(k^{(i_1,l_{x3})}_{j_{x3}}, \mathrm{ID}_b)$. Node a can calculate these keys by using the generation keys of these keys $k^{i_1}_{j_1}, \ldots, k^{i_1}_{j_{x3}}$ and the same method as the previous.

As a result, if the number of shared keys is larger than 0, that is, $x1 + x2 + x3 \geq 1$, the shared key between them is hashed by all common keys.

5. Performance Analysis and Simulation

In this section, we will analyze and simulate the performances of our scheme, including deployment model, local connectivity, communication overhead, and network resilience against node capture.

In the following experiments, the involved main parameters subsequent are defined as follows.

(1) We consider a SN deployed over fields of 475 m by 520 m.

(2) The area is divided into a hexagon and *len* is 50.

(3) The center of each cell is the deployment point (see Figure 1).

(4) The number of nodes in a $G_{(r,c)}$ is 50 $(N = 50)$.

(5) The wireless communication range for a node is 40 m.

(6) We assume that node deployment follows a two-dimensional Gaussian distribution [27], and its standard deviation is $\sigma = 40$.

(7) We assume that node deployment includes 5 phases. The value of ρ_0 is 0.7.

(8) The number of key pool of a cell, namely, m, is 175, and the length of a forward key chain is 30 $(L = 30)$.

5.1. ρ_0. In the capture model, when an attacker locates in a cell, he can capture nodes around it. In this paper, it is supposed that compromised nodes can be identified by using schemes proposed by some scholars, such as [28]. But how

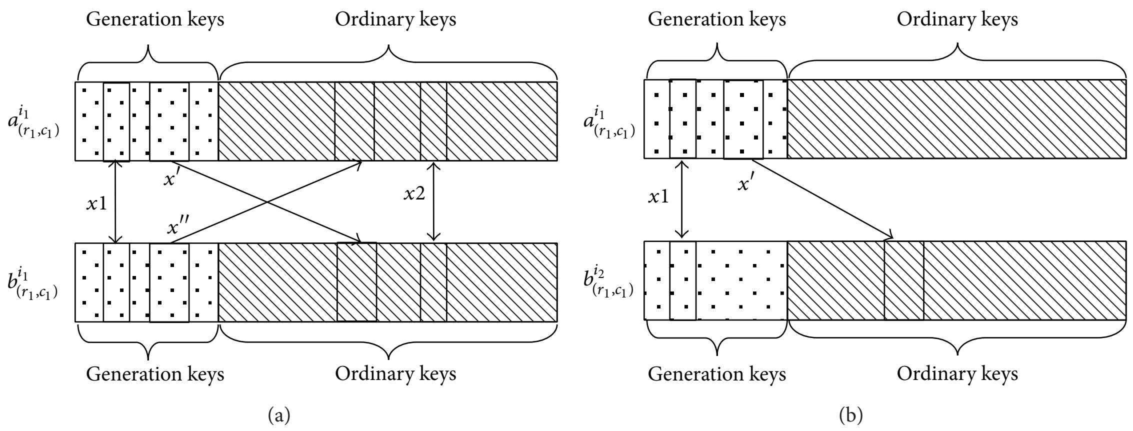


FIGURE 4: The common key between two sensors, where $i_1 > i_2$ in (b).

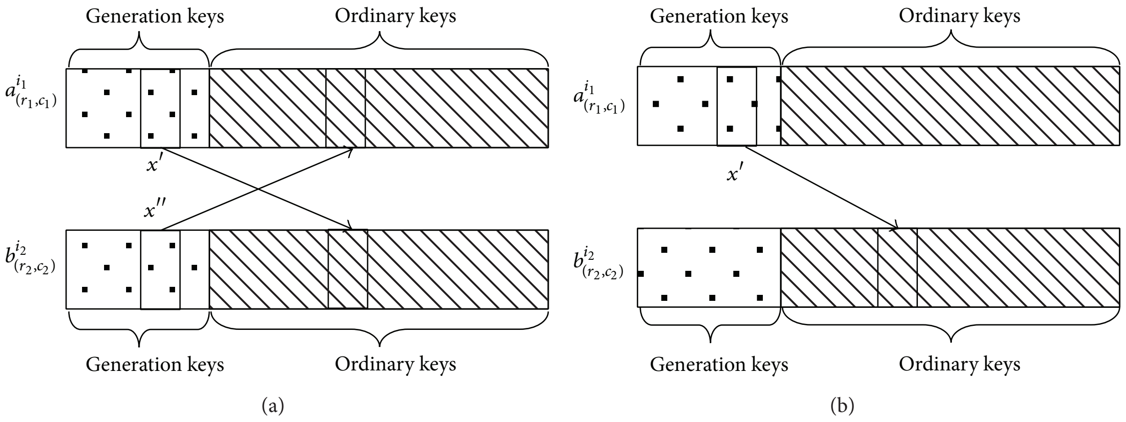


FIGURE 5: The common key between two sensors, where $i_1 > i_2$ in (b).

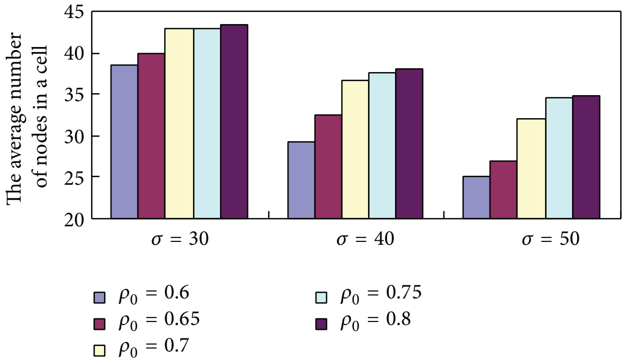


FIGURE 6: The average number of uncaptured nodes in a cell as a function of σ and ρ_0.

can we determine the value of ρ_0 when we do not know nodes' location information? The value of ρ_0 and the actual number of uncaptured nodes in a cell are closely related. For example, in the first phase, if nodes in cell $(0,0)$ is 50, 50% of these nodes are captured, and the number of these captured nodes whose deployment points reside in cells $(0,0)$, $(0,1)$, $(1,0)$, $(2,0)$, $(1,1)$, and $(0,2)$ is 16, 3, 2, 2, 1, and 1, respectively. Here, it is supposed that these captured nodes can be identified. Therefore, $\rho_{(0,0)}$, $\rho_{(0,1)}$, $\rho_{(1,0)}$, $\rho_{(2,0)}$, $\rho_{(1,1)}$, and $\rho_{(0,2)}$ are equal to 0.68, 0.94, 0.96, 0.96, 0.98, and 0.98, respectively. When $\rho_0 = 0.6$, no new nodes are needed to be added to these cells, which will cause that only a few uncaptured nodes in cell $(0,0)$ are working in a long time.

Figure 6 shows that the relation between the number of uncaptured nodes in a cell and the standard deviation of two-dimension normal distributions and ρ_0. In our simulations, the number of nodes in each cell is about 48, and if a cell is compromised, about 50% nodes in the cell are captured. When the ratio of nodes uncaptured in the set $\text{SN}_{(r,c)}$ is less than ρ_0, new nodes are needed to be added to cell (r,c). The larger the σ, the smaller the number of new nodes which actually reside in cell (r,c). Therefore, when σ increases, the number of nodes captured in cell (r,c) which come from $G_{(r,c)}$ decreases. To ensure that nodes uncaptured in each cell are many, we must increase ρ_0. For example, after nodes are captured and new nodes are added to the network, to ensure that the average number of nodes in the captured cells is larger than 35, when $\sigma = 30$ and $\sigma = 40$, the value of ρ_0 should be set to 0.6 and 0.7, respectively. However, when $\sigma = 50$, even if ρ_0 equals 0.8, the above goal cannot be achieved. In addition, the frequency of adding new nodes to the network increases with the increase of ρ_0. Therefore, the value of ρ_0 should be set in accordance with the specific condition. For example, when $\sigma = 30$ and $\sigma = 40$, we can set ρ_0 to 0.7.

5.2. Local Connectivity. For multiple deployment sensor networks, local connectivity is not only affected by the key predistribution method but also affected by the deployment model and the capture model. In this paper, only the analysis that local connectivity is affected by the key predistribution method is presented, that is, the probability $P_{(a,b)}$ of shared key between nodes a and b.

If $r_1 = r_2$ and $c_1 = c_2$, when $T_b = T_a$, the probability $P_{(a,b)}$ can be calculated as follows: a and b have $\binom{m/7}{t1}^6 \cdot \binom{m/7}{t3}^6 \cdot \binom{m/7}{t2+t3} \cdot \binom{t2+t3}{t3} \cdot \binom{L}{1}^{6t1+t2}$ different ways of picking their $6t1 + t2 + 7t3$ keys from the ordinary key pools and the generation key pools. It is supposed that a and b have x two-dimensional backward key chains in common. The number of ways to pick $x1$ generation keys from generation key pools of neighbor cells and $x2 + x3$ ordinary keys from ordinary key pools $Pc^i_{(r_1,c_1)_s}$ (see Section 4.3.2) can be calculated as follows:

$$P_x = \sum_{x=1}^{6t1+t2+7t3} \sum_{\substack{x1=\sum_{i=0}^{6} y_i,\\ x2=\sum_{i=0}^{6} y_i'}} PS_0 \cdot \prod_{i=1}^{6} \binom{\frac{m}{7}}{y_i} \cdot \binom{\frac{m}{7} - y_i}{2\left(t3 - y_i\right)} \cdot \binom{2\left(t3 - y_i\right)}{t3 - y_i} \cdot \prod_{i=1}^{6} \sum_{x_5=0}^{m/7-y_i'} \binom{\frac{m}{7}}{y_i' + x_5} \cdot \binom{y_i' + x_5}{x_5} \cdot \binom{\frac{m}{7} - y_i' - x_5}{2\left(t1 - y_i' - x_5\right)} \cdot \binom{2\left(t1 - y_i' - x_5\right)}{t1 - y_i' - x_5} \cdot \binom{L}{1}^{2t1-y_i'-2x_5} \cdot \binom{L-1}{1}^{x_5}, \tag{4}$$

where PS_0 can be calculated as

$$PS_0 = \sum_{x3=x'+x''=1}^{t3-y_0} \sum_{x_4=0}^{2t2-x3} \binom{\frac{m}{7}}{x3 + y_0 + y_0' + x_4} \cdot \binom{y_0' + x_4}{x_4} \cdot \binom{\frac{m}{7} - x3 - y_0 - y_0' - x_4}{2\left(t2 + t3 - x3 - y_0 - y_0' - x_4\right)} \cdot \binom{2\left(t2 + t3 - x3 - y_0 - y_0'\right)}{t2 + t3 - x3 - y_0 - y_0'} \cdot \binom{t2 + t3 - x3 - y_0 - y_0' - x_4}{t3 - x' - y_0} \cdot \binom{t2 + t3 - x3 - y_0 - y_0' - x_4}{t3 - x'' - y_0} \cdot \binom{L}{1}^{2t2-y_0'-2x_4} \cdot \binom{L-1}{1}^{x_4}. \tag{5}$$

When $T_a \neq T_b$ (without loss of generality, it is supposed that $T_b < T_a$), $x'' = x2 = 0$. The number of ways to pick $x1$ generation keys from generation key pools of neighbor cells and the $x3$ ordinary keys from ordinary key pools $Pc^i_{(r1,c1)_s}$ (see Section 4.3.2) can be calculated as follows:

$$P_x' = \sum_{x=1}^{6t1+t2} \sum_{\substack{x1=\sum_{i=0}^{6} y_i,\\ x2=\sum_{i=0}^{6} y_i'}} PS_0' \cdot \prod_{i=1}^{6} \cdot \binom{\frac{m}{7} - y_i}{2\left(t3 - y_i\right)} \cdot \binom{2\left(t3 - y_i\right)}{t3 - y_i} \cdot \prod_{i=1}^{6} \binom{\frac{m}{7}}{y_i'} \cdot \binom{\frac{m}{7} - y_i'}{2\left(t1 - y_i'\right)} \cdot \binom{2\left(t1 - y_i'\right)}{t1 - y_i'} \cdot \binom{L}{1}^{12t1}, \tag{6}$$

where PS_0' can be calculated as

$$PS_0' = \sum_{x3=x'=1}^{t3-y_0} \binom{\frac{m}{7}}{x3 + y_0} \cdot \binom{x3 + y_0}{x3} \cdot \binom{\frac{m}{7} - x3 - y_0}{2\left(t2 + t3 - x3 - y_0\right)} \cdot \binom{2\left(t2 + t3 - x3 - y_0\right)}{t2 + t3 - x3 - y_0} \cdot \binom{t2 + t3 - x3 - y_0}{t3 - x' - y_0} \cdot \binom{t2 + t3 - x3 - y_0}{t3 - y_0} \cdot \binom{L}{1}^{2t2}. \tag{7}$$

Hence, if $r_1 = r_2$ and $c_1 = c_2$, we have

$$P_{(a,b)} = \begin{cases} \dfrac{P_x}{\left(\binom{m/7}{t1}^6 \cdot \binom{m/7}{t3}^6 \cdot \binom{m/7}{t2+t3} \cdot \binom{t2+t3}{t3} \cdot \binom{L}{1}^{6t1+t2}\right)^2} & \text{when } T_a = T_b, \quad \text{where } x = x1 + x2 + x3 \\ \dfrac{P_x'}{\left(\binom{m/7}{t1}^6 \cdot \binom{m/7}{t3}^6 \cdot \binom{m/7}{t2+t3} \cdot \binom{t2+t3}{t3} \cdot \binom{L}{1}^{6t1+t2}\right)^2} & \text{when } T_a \neq T_b, \quad \text{where } x = x1 + x3. \end{cases} \tag{8}$$

If cells (r_1, c_1) and (r_2, c_2) are neighbor cells, $x1 = x2 = 0$. When $T_b = T_a$, a and b have $\binom{m/7}{t1} \cdot \binom{m/7}{t3} \cdot \binom{L}{1}^{t1}$ different ways of picking $t1$ ordinary keys from the ordinary key pool and $t3$ ordinary keys from the generation key pool. The number of ways to pick the x ($x = x' + x''$) common keys can be calculated as follows:

$$P1_x = \binom{\frac{m}{7}}{t1} \cdot \binom{\frac{m}{7}}{t1} \cdot \binom{L}{1}^{2t1} \cdot \sum_{x=x_3=1}^{t3} \sum_{x'+x''=x_3} \binom{t1}{x'} \cdot \binom{t1}{x''} \cdot \binom{\frac{m}{7}-t1}{t3-x''} \cdot \binom{\frac{m}{7}-t1}{t3-x'}. \tag{9}$$

When $T_a \neq T_b$ (without loss of generality, it is supposed that $T_b < T_a$), $x1 = x2 = x'' = 0$. The number of ways to pick the x ($x = x'$) keys can be calculated as follows:

$$P1'_x = \binom{\frac{m}{7}}{t1} \cdot \binom{\frac{m}{7}}{t1} \cdot \binom{\frac{m}{7}}{t3} \cdot \binom{L}{1}^{2t1} \cdot \sum_{x_3=1}^{t3} \binom{t1}{x3} \cdot \binom{\frac{m}{7}-t1}{t3-x3}. \tag{10}$$

Hence, if cells (r_1, c_1) and (r_2, c_2) are neighbor cells, we have

$$P_{(a,b)} = \begin{cases} \dfrac{P1_x}{\left(\binom{m/7}{t1} \cdot \binom{m/7}{t3} \cdot \binom{L}{1}^{t1}\right)^2} \\ = \dfrac{\sum_{x=x_3=1}^{t3} \sum_{x'+x''=x_3} \binom{t1}{x'} \cdot \binom{t1}{x''} \cdot \binom{m/7-t1}{t3-x''} \cdot \binom{m/7-t1}{t3-x'}}{\binom{m/7}{t3}^2} \\ \qquad \text{when } T_a = T_b \\ \dfrac{P1'_x}{\left(\binom{m/7}{t1} \cdot \binom{m/7}{t3} \cdot \binom{L}{1}^{t1}\right)^2} = \dfrac{\sum_{x3=1}^{t3} \binom{t1}{x3} \cdot \binom{m/7-t1}{t3-x3}}{\binom{m/7}{t3}} \\ \qquad \text{when } T_a \neq T_b. \end{cases} \tag{11}$$

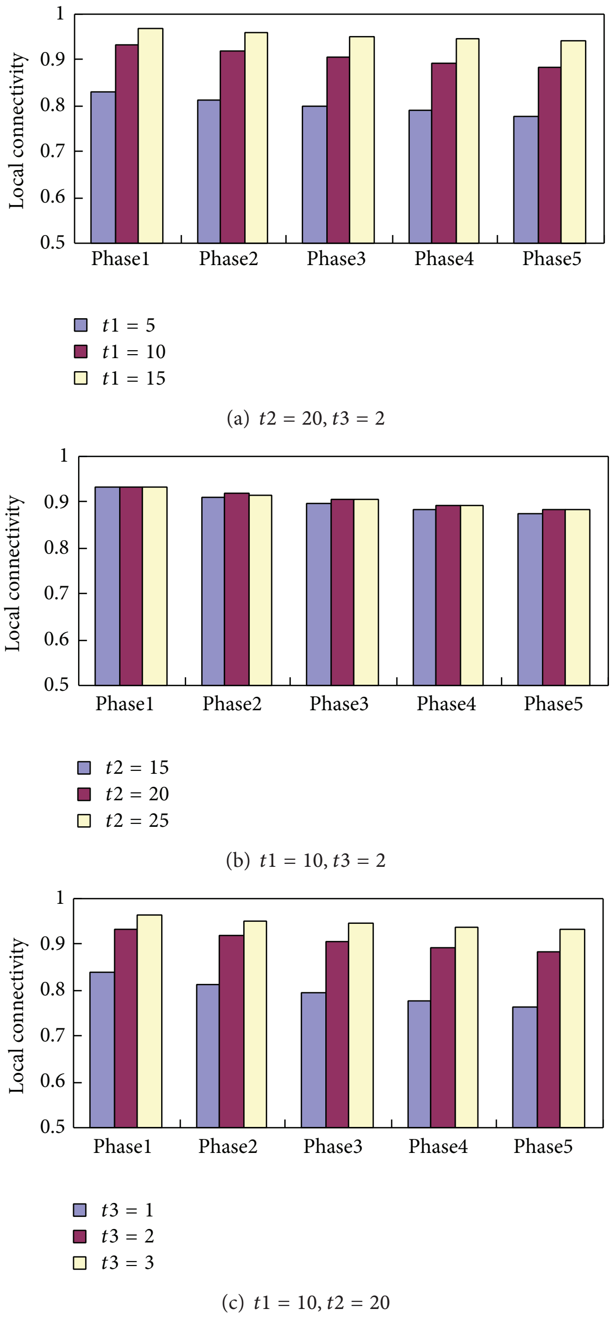
If cell (r_1, c_1) and cell (r_2, c_2) are not neighbor cells, the probability $P_{(a,b)}$ is equal to 0.

Figure 7 shows that local connectivity of our scheme is high. For example, when $t1 = 10$, $t2 = 20$, and $t3 = 2$, the local connectivity in the first phase is 0.936. And in this case, the number of keys predistributed to a node is $10 \times 6 + 20 + 2 \times 7 = 94$ only. In addition, Figure 7 shows that the larger the value of $t1$ and $t3$, the higher the local connectivity. For example, the value of $t1$ increases from 5 to 15 and $t3$ increases from 1 to 3; local connectivity increases by 0.135 and 0.117, respectively. However, in this case, for a node, the storage overheads increase by 60 and 14, respectively. As a result, we can have a conclusion that the parameter $t3$ has a great influence on local connectivity. The larger the value of $t3$ is, the higher the local connectivity is. However, the larger the value of $t3$, the more keys compromised when nodes are captured. If a node is captured, the number of keys compromised can be calculated by $6 \times t1 + t2 + 7 \times t3 \times i_a \times L$, where i_a represents the deployment phase of node a. When i_a and L are 5 and 30, respectively, $t3$ increases from 1 to 3; the number of keys compromised increases by $7 \times 2 \times 5 \times 30 = 2100 \gg 6 \times 10 = 60$. Therefore, concerning safety, $t3$ should be as small as possible. On the other hand, Figure 7 shows that the larger the number of deployment phases, the smaller the local connectivity. However, after multiple deployments, local connectivity can keep basically stable. For example, from the 4th phase to the 5th phase, local connectivity decreases by less than 0.01.

5.3. Communication Overhead. If direct key establishment fails, two sensor nodes need to start on path-key establishment phase to establish a pairwise key with the help of other sensor nodes. To establish a pairwise key with node b, node a needs to find a sequence of nodes between itself and node b such that any two adjacent nodes in this sequence can establish a direct key. For the sake of presentation, we call such a sequence of nodes a *key path*.

In this section, we investigate the number of hops required on this path for various parameters of our scheme. Let ph(h) be the probability that the smallest number of hops needed to connect two neighboring nodes is h. Obviously, ph(1) is local connectivity.

In our scheme, after the 5th deployment, the local connectivity keeps basically stable. So, we plot the values of ph(1), ph(2), ph(3), and ph(4) of the four phases (see Figure 8). From the figure, we can observe that ph(1) + ph(2) $\approx$ 1 (i.e., the probability that at most 2 hops are required is essentially 1).



(a) $t2 = 20, t3 = 2$

(b) $t1 = 10, t3 = 2$

(c) $t1 = 10, t2 = 20$

Figure 7: Local connectivity as a function of various parameters.

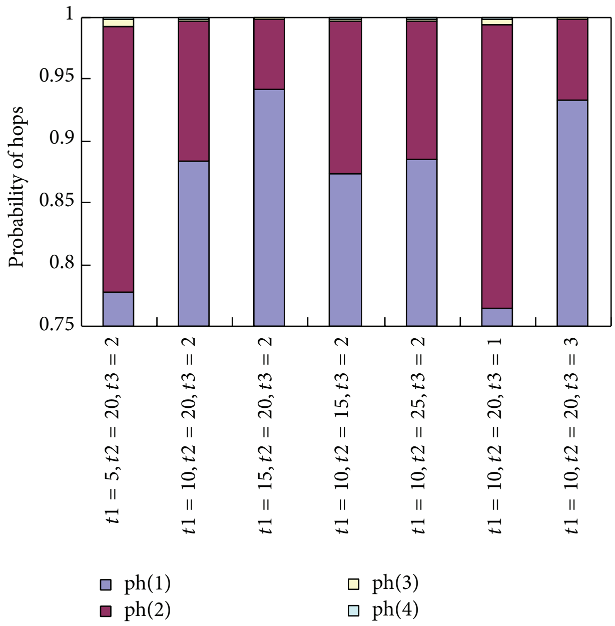


Figure 8: Distribution of the number of hops required to connect neighbors.

5.4. Comparisons. In this section, performance and security between our scheme and YG scheme [17] and ESPK scheme [20], are compared. For the sake of fairness, in YG and ESPK, the method for processing keys is same as our scheme. In this simulation, $t1$, $t2$, and $t3$ of our scheme is 10, 20, and 2, respectively. The predistribution keys of YG and ESPK scheme is same as our scheme. Other parameters are same as Section 5.

5.4.1. Local Connectivity. With the same storage overhead, Figure 9 shows that, in each deployment phase, local connectivity of our scheme is higher than YG scheme and *ESPK scheme*. In *ESPK scheme*, nodes which come from different deployment phase but reside in the same cell cannot communicate with each other. Its local connectivity is the lowest. Fox example, in the fifth phase, local connectivity of our scheme, YG scheme, and ESPK scheme is 0.88, 0.86, and 0.46, respectively.

5.4.2. Global Connectivity. If local connectivity is less than 1, nodes in SNs may be divided into one or more isolated components. Any two nodes in an isolated component can securely communicate with each other directly or indirectly (Figure 10). Global connectivity refers to the ratio of the number of nodes in the largest isolated component to the size of the whole network. If the ratio equals 98%, it means that 98% of the sensor nodes are connected securely and the remaining 2% are unreachable from the largest isolated component. So, global connectivity metric indicates the percentage of nodes that are wasted because of their unreachability.

In this work, we use simulation to estimate it. In *ESPK scheme*, nodes which reside in the same cell but are deployed in different deployment phases can establish shared keys only by using path keys. For a node, if it cannot find a path to establish shared key with neighbouring nodes, then it is an isolated node. Therefore, the global connectivity of the scheme is the lowest. Fox example, in the fifth phase, global connectivity of our scheme, YG scheme, and ESPK scheme is 0.99955, 0.99941, and 0.94499, respectively.

5.4.3. Resilience. A scheme's resilience toward node capture is evaluated by estimating the fraction of total network communications that are compromised by a capture of x nodes *not including* the communications in which the compromised nodes are directly involved.

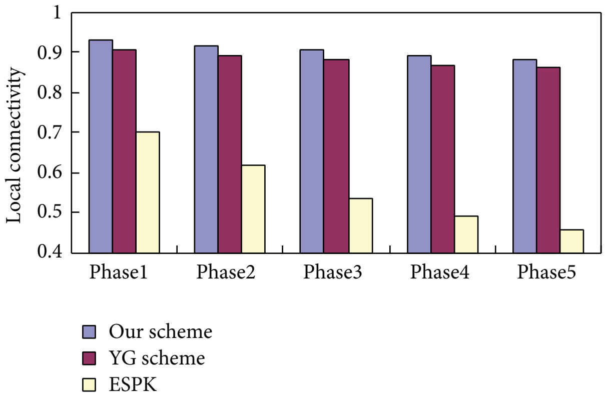


Figure 9: Comparing the local connectivity of our, YG, and ESPK scheme.

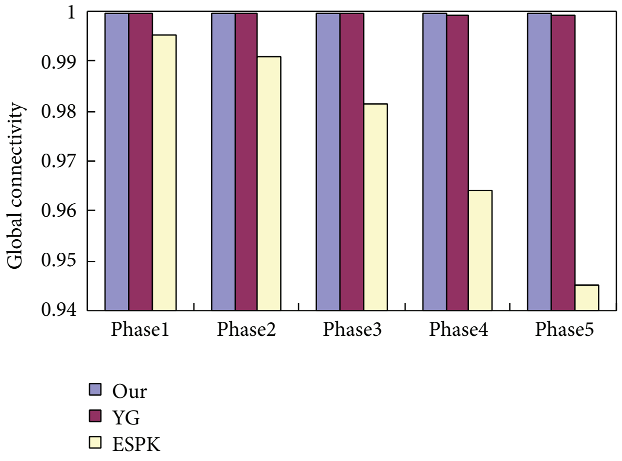


Figure 10: Comparing the global connectivity of our, YG, and ESPK scheme.

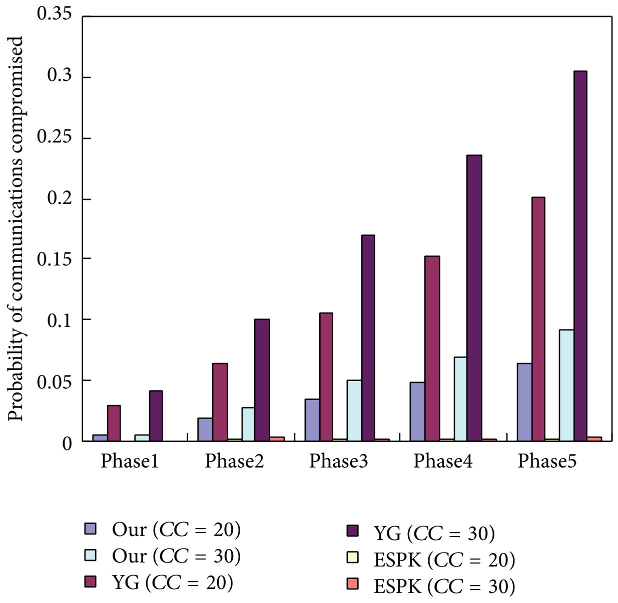


Figure 11: Comparing the resilience of our, YG, and ESPK scheme. Where CC is the number of nodes captured by an attacker during the period of the direct key establishment phase.

We conducted simulation tests on network resilience against node capture of the above three schemes. In our simulations, it is supposed that only a few nodes are compromised during the shared-key discovery phase. In ESPK scheme, the key pools of nodes from different deployment phase are different. Therefore, its network resilience against node capture is the best. In YG scheme, the key pool is fixed. Therefore, increases in the number of captured nodes will diminish network resilience. For example, when $CC = 30$, Figure 11 shows the probability that a shared key is compromised in the first phase and the 5th phase is 0.04 and 0.31, respectively. In our scheme, the subkey pool of the ith phase and the i'th phase is disjoint, that is, $P^{i}_{(r,c)} \cap P^{i'}_{(r,c)} = \phi$ ($i \neq i'$). Therefore, compared with YG scheme, our scheme can improve the performance in network resilience against node capture attacks. For example, when $CC = 30$, in our scheme, the probability that shared keys are compromised in the 5th phase is 0.09.

6. Conclusion and Future Work

In this paper, we proposed a new deployment model for multiple deployment sensor networks, based on which a new key management scheme is further presented. We conducted a comprehensive study on connectivity, network resilience of our scheme. The results showed that our scheme can significantly improve network resilience over the YG scheme [17]. Compared with the ESPK scheme [20], our scheme can significantly improve its local connectivity and global connectivity, although the resilience of our scheme is poorer than that of the scheme. We have presented both the analytical and numerical results. In our future work, we will study different attack models and the accuracy how attack model affects the results.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is partially supported by the National Natural Science Foundations of China (61232001 and 61173169), the Research Foundation of Education Committee of Hunan Province, China (13B057, 09A046 and 10A062), the Hunan Provincial Natural Science Foundation of China (12JJ2040), the Construct Program of the Key Discipline in Hunan Province, China, the Aid program for Science and Technology Innovative Research Team in Higher Educational Institute of Hunan Province, and the Construct Program of the Key Discipline in Hunan University of Humanities, Science and Technology.

References

[1] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[2] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[3] L. Wei, H. Zhu, Z. Cao et al., "Security and privacy for storage and computation in cloud computing," *Information Sciences*, vol. 258, pp. 371–386, 2014.

[4] L. Wei, H. Zhu, Z. Cao, W. Jia, and A. V. Vasilakos, "SecCloud: bridging secure storage and computation in cloud," in *Proceedings of the IEEE 30th International Conference on Distributed Computing Systems Workshops (ICDCSW '10)*, pp. 52–61, June 2010.

[5] Z. Zhang, H. Wang, A. V. Vasilakos, and H. Fang, "ECG-cryptography and authentication in body area networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1070–1078, 2012.

[6] H. Yang, Y. Zhang, Y. Zhou, X. Fu, H. Liu, and V. V. Athanasios, "Provably secure three-party authenticated key agreement protocol using smart cards," *Computer Networks*, vol. 58, no. 1, pp. 29–38, 2014.

[7] K. Piotrowski, P. Langendoerfer, and S. Peter, "How public key cryptography influences wireless sensor node lifetime," in *Proceedings of the 4th ACM Workshop on Security of ad hoc and Sensor Networks (SASN '06)*, pp. 169–176, October 2006.

[8] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 41–47, November 2002.

[9] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the IEEE Symposium on Security And Privacy*, pp. 197–213, May 2003.

[10] W. Bechkit, "New key management schemes for resource constrained wireless sensor networks," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '11)*, June 2011.

[11] D. Liu, P. Ning, and L. I. Rongfang, "Establishing pairwise keys in distributed sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 1, pp. 41–77, 2005.

[12] M. Ehdaie, N. Alexiou, M. Ahmadian, M. R. Aref, and P. Papadimitratos, "Key splitting for random key distribution schemes," in *Proceedings of the 20th IEEE International Conference on Network Protocols (ICNP '12)*, pp. 1–6, November 2012.

[13] P. Papadimitratos and J. Deng, "Stealthy pre-attacks against random key pre-distribution security," in *Proceedings of the IEEE International Conference on Communications (ICC '12)*, pp. 955–959, June 2012.

[14] J. F. Kazienko, I. G. Ribeiro, I. M. Moraes, and C. V. N. Albuquerque, "SENSORLock: a lightweight key management scheme for wireless sensor networks," *Security and Communication Networks*, vol. 6, no. 10, pp. 1198–1210, 2013.

[15] J. Ibriq and I. Mahgoub, "HIKES: hierarchical key establishment scheme for wireless sensor networks," *International Journal of Communication Systems*, 2012.

[16] W. Du, J. Deng, and Y. S. Han, "A key predistribution scheme for sensor networks using deployment knowledge," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 1, pp. 62–77, 2006.

[17] Z. Yu and Y. Guan, "A key management scheme using deployment knowledge for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 10, pp. 1411–1425, 2008.

[18] A. Fanian, M. Berenjkoub, H. Saidi, and T. Aaron Gulliver, "A high performance and intrinsically secure key establishment protocol for wireless sensor networks," *Computer Networks*, vol. 55, no. 8, pp. 1849–1863, 2011.

[19] N. T. T. Huyen, M. Jo, T.-D. Nguyen, and E.-N. Huh, "A beneficial analysis of deployment knowledge for key distribution in wireless sensor networks," *Security and Communication Networks*, vol. 5, no. 5, pp. 485–495, 2012.

[20] 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.

[21] S. Li, B. Zhou, J. Dai, and X. Sun, "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.

[22] B. Zhou, J. Wang, S. Li, Y. Cheng, and J. Wu, "A continuous secure scheme in static heterogeneous sensor networks," *IEEE Communications Letters*, vol. 17, no. 9, pp. 1868–1871, 2013.

[23] B. Zhou, S. Li, J. Wang, S. Yang, and J. Dai, "A pairwise key establishment scheme for multiple deployment sensor networks," *International Journal of Network Security*, vol. 16, no. 3, pp. 221–228, 2014.

[24] D. He, C. Chen, S. Chan, J. Bu, and A. V. Vasilakos, "A distributed trust evaluation model and its application scenarios for medical sensor networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 6, pp. 1164–1175, 2012.

[25] X. Ma, H. Song, J. Wang, J. Gao, and G. Min, "A novel verification scheme for fine-grained Top-k queries in two-tiered sensor networks," *Wireless Personal Communications*, vol. 75, no. 3, pp. 1809–1826, 2014.

[26] J. Wang, Z. Liu, S. Zhang, and X. Zhang, "Defending collaborative false data injection attacks in wireless sensor networks," *Information Sciences*, vol. 254, no. 1, pp. 39–53, 2014.

[27] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, Addison-Wesley, Reading, Mass, USA, second edition, 1994.

[28] Q. Zhang, T. Yu, and P. Ning, "A framework for identifying compromised nodes in wireless sensor networks," *ACM Transactions on Information and System Security*, vol. 11, no. 3, pp. 1–37, 2008.

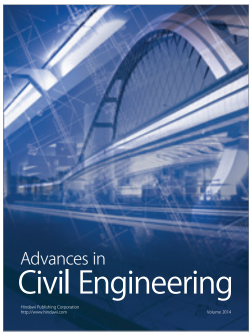
Journal of
Engineering

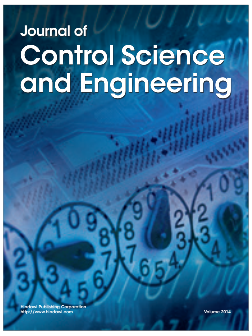
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

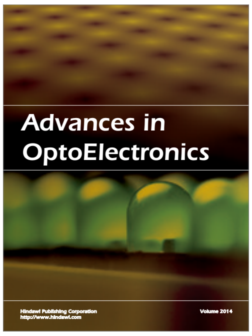
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

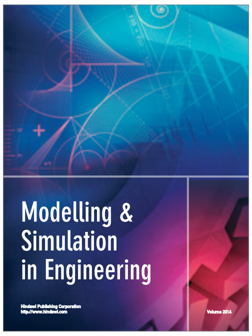
Journal of
Robotics

Journal of
Electrical and Computer Engineering

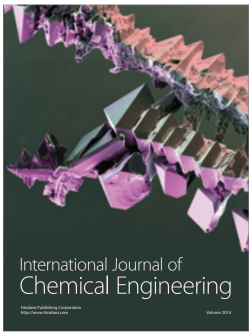

Advances in
OptoElectronics

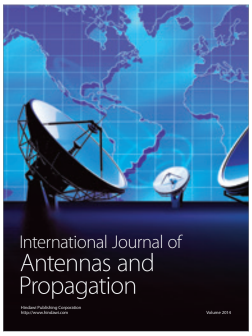
VLSI Design

International Journal of
Navigation and Observation

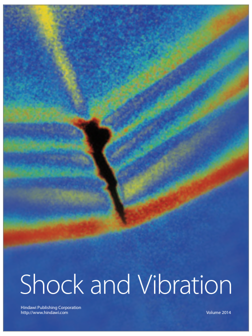

Modelling & Simulation in Engineering

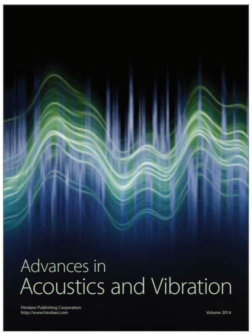
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration