Received 31 March 2013; revised 29 August 2013; accepted 6 October 2013. Date of publication 30 October 2013;
date of current version 21 January 2014.

Digital Object Identifier 10.1109/TETC.2013.2287196

A Matrix-Based Pairwise Key Establishment Scheme for Wireless Mesh Networks Using Pre Deployment Knowledge

YUEXIN ZHANG[1], LI XU[1], YANG XIANG[2], AND XINYI HUANG[1]

[1]Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350108, China

[2]School of Information Technology, Deakin University, Burwood 3125, Australia

CORRESPONDING AUTHOR: L. XU (xuli@fjnu.edu.cn)

This work was supported in part by National Natural Science Foundation of China (Grant NO.61072080 and NO.61202450), Ph.D. Programs Foundation of Ministry of Education of China (Grant NO. 20123503120001), Natural Science Foundation of Fujian Province (NO.2013J01222), Department of Education, Fujian Province, A-Class Project (Grant NO.JA12076), Distinguished Young Scholars Fund of Department of Education, Fujian Province, China (JA13062), and the development project of Fujian provincial strategic emerging industries technologies: Key technologies in development of next generation Integrated High Performance Gateway, Fujian development and reform commission high-technical [2013]266.

This paper is an extended version of "Matrix-based Pairwise Key Establishment in Wireless Mesh Networks Using Deployment Knowledge", accepted by IEEE ICC 2013-Ad Hoc and Sensor Networking Symposium, June 9–13, 2013, Budapest, Hungary.

ABSTRACT Due to the nature of wireless transmission, communication in wireless mesh networks (WMNs) is vulnerable to many adversarial activities including eavesdropping. Pairwise key establishment is one of the fundamental issues in securing WMNs. This paper presents a new matrix based pairwise key establishment scheme. Mesh client in our scheme only needs to prestore a key seed, which can be used to generate a column of secret matrix. It can establish pairwise keys with other clients after mesh routers broadcast public matrices. Our scheme is motivated by the fact that in WMNs, mesh routers are more powerful than mesh clients, both in computation and communication. Besides, we employ the pre deployment knowledge to reduce the computational cost of mesh clients. Security and complexity analysis show that the new scheme possesses several desirable features: 1) neighbor mesh clients can directly establish pairwise keys; 2) the new scheme is updatable, scalable, and robust against node capture attacks; and 3) communication and storage costs at mesh clients are significantly reduced.

INDEX TERMS Pairwise key, matrix, deployment knowledge, wireless mesh networks, wireless sensor networks.

I. INTRODUCTION

Cyber-Physical Systems (CPS), which bridge the cyber-world of computing and communications with the physical world, are expected to change the way of interacting with and controlling the physical world around us. It is typically designed, not as a network of standalone devices, but as interacting elements with physical input and output. It's applications, such as transportation vehicles and intelligent highways, robotic systems, factory automation, and smart building, do have enormous societal impacts and economic benefits. As a representative network type of CPS, wireless mesh networks (hereinafter, WMNs) have attracted great attention from academia and industry. WMNs are dynamically self-organized and self-configured, with the nodes in the network automatically establishing an Ad Hoc network and maintaining mesh connectivity [1]. A WMN consists of two types of nodes: mesh clients and mesh routers. Mesh clients are either stationary or mobile devices, and mesh routers form the mesh backbone for mesh clients. Each node (including mesh clients and mesh routers) in WMNs operates as a host and as a router, but mesh routers are not as power-constrained as mesh clients in WMNs and thus can accommodate more resource-intensive tasks. As shown in Fig. 1, the gateway/bridge functionalities of mesh routers enable the integration of WMNs with other networks, including vehicular networks, Ad Hoc networks, Wi-Fi, cellular networks, wireless sensor networks [2].

The nature of wireless transmission is the major factor contributing to the vulnerability of WMNs under a variety of malicious cyber attacks. For example, adversaries can eavesdrop, interrupt and modify transmission, impersonate

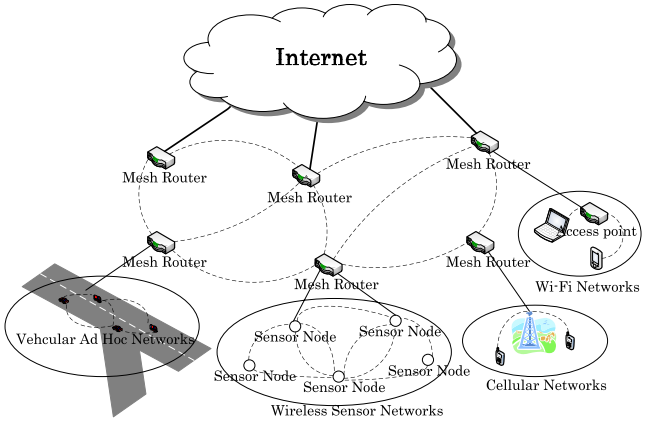


FIGURE 1. Network architecture: Infrastructure/backbone WMNs presented in Akyildiz et at. scheme [2].

legitimate nodes, and even capture nodes in WMNs. Many security mechanisms have been proposed to countermeasure potential attacks in WMNs, and a non-exhaustive list includes location technology [3], [4], intrusion detection technology [5], [6], secure routing technology [7], [8] and key management technology [9]–[23].

As a fundamental problem, key management has been both extensively and intensively studied in CPS and other similar situations. Taking wireless sensor networks (which are often integrated with WMNs) for example, asymmetric key cryptographic algorithms are generally considered infeasible for computing and communicating between energy-constrained sensor nodes [9]. Though such a constraint has been partially alleviated with the development of modern technology, it is still a fact that sensor nodes are not able to afford frequent asymmetric cryptographic operations. Other mechanisms, such as Kerberos, cannot be directly applied to sensor networks due to the lack of trusted infrastructure [10].

According to its characteristics, key management can be classified by Self-enforcing Schemes, Arbitrated Keying Schemes and Key Pre-distribution Schemes [11], where Key Pre-distribution Schemes (hereinafter, KPS) is the focus of this paper. In KPS, a key management authority (also known as key distribution center) loads keys into nodes prior to deployment, then neighbor nodes can establish secure communication keys using their pre-loaded keys. Arguably the most straightforward KPS is to equip all nodes with a common master key, and any node can negotiate a session key with each other using the master key after deployment. However, an inherent weakness of this approach is that all communications within the network will become insecure if one node is captured by the adversary. Here, we assume that node-capture provides adversaries with all sensitive data stored in that node. Note that tamper-resistance can increase the difficulty of data retrieving, it is more preferable that KPS remains a certain level of security if tamper-resistance fails to achieve its purpose due to numerous known attacks (e.g., microprobing and power analysis) and other unknown attacks in the short future.

When wireless sensor networks are integrated with WMNs, as shown in Fig. 1, there are a large number of energy-constrained sensor nodes (i.e., mesh clients) in WMNs. In some applications, these power-constrained nodes need to establish communication keys in a short time after deployment. In this paper, we propose a new matrix-based pairwise key establishment scheme to meet the need of such applications.

Our Contribution. The major contribution of this paper is a new design of pairwise key establishment for network type of CPS: WMNs.

In WMNs, sensor nodes are power-constrained but mesh routers are much more powerful, both in computation and communication. Such a nature of heterogeneity makes it feasible for sensor nodes to establish pairwise keys by delegating costly operations to mesh routers. In this paper, we modified Blom's scheme and showed a new design of matrix G. In our scheme, G is a secret matrix and an independent key seed s_i is given to the i^{th} node, which can only generate the i^{th} column of G but does not have any other information about G. Furthermore, we employ the pre deployment knowledge to reduce the computational cost of sensor nodes.

Security and complexity analysis indicates that our scheme possesses the following properties:

1) Neighbor sensor nodes can directly establish pairwise keys;
2) The scheme is updatable, scalable and robust against node capture attacks;
3) Our scheme has significant advantages in terms of storage cost and communication cost at sensor nodes; and
4) While our scheme is specifically designed for key establishment in WMNs, it is also applicable in other CPS situations with the similar feature of heterogeneity.

Organization of This Paper

The remainder of this paper is organized as follows. In Section II, we overviews related works in wireless sensor networks. The proposed scheme is described in Section III, and its security and performance analysis is given in Section IV. Section V concludes this paper.

II. RELATED WORK

A widely accepted requirement of robust KPS is that an adversary, after capturing several nodes, should be difficult to derive the communication keys of other nodes or disrupt the entire networks. For this purpose, a pairwise key between two nodes is necessary. A naïve way of designing robust KPS is to pre-distribute each sensor node with $N-1$ keys, where N is the total number of nodes and each one of $N-1$ keys is shared with each one of $N-1$ nodes. After deployment, any pair of nodes will share a pairwise key. Adversaries with captured nodes will have keys associated with compromised nodes only, but not those among un-compromised nodes. Such a mechanism provides a high level of robustness but a low level of scalability: the performance of key distribution phase will be time, computation and storage consuming when N is large,

and pairwise key establishment between any current node and newly added one needs a system-wide update.

A. CLASSICAL KEY PRE-DISTRIBUTION SCHEMES

A number of KPS have been proposed to provide different tradeoffs between robustness and scalability. The scheme by Eschenauer and Gligor [12] is based on symmetric encryption. In key pre-distribution phase, system authority pre-distributes each node with m keys (called key rings) and key identifiers from a large key pool P before nodes deployment. Neighbor nodes then can broadcast key identifiers and successfully establish session keys using their shared pre-distributed keys. The major drawback of Eschenauer-Gligor's scheme is that different pairs of nodes may share a same session key, and as a result there is a risk that the communication among un-compromised nodes may become insecure after the compromise of other nodes. This security flaw is partly improved by Chan *et al.* by extending Eschenauer and Gligor's idea to a q-composite random key pre-distribution scheme [13], where neighbor nodes can establish a session key only if they share t ($t \geq q$) common keys. Nevertheless, there still exist constrains between the key pool size P, the security parameter q and m.

B. MATRIX-BASED SCHEMES

Matrix is a commonly used mathematic tool for pairwise key establishment. The pioneering work is introduced by Blom, who proposed a KPS allowing any pair of nodes to establish pairwise key directly [14]. In Blom's scheme, there exists a $(\lambda+1)\times N$ public matrix G over a finite field $GF(q)$, where N is the size of network nodes and $q > N$. System authority generates a random $(\lambda+1)\times(\lambda+1)$ symmetric matrix D as the secret key information over $GF(q)$ in key pre-distribution phase and computes an $N\times(\lambda+1)$ matrix $A = (D\cdot G)^T$, where $(D\cdot G)^T$ is the transpose of $D\cdot G$. Then system authority loads the k^{th} row of matrix A and k^{th} column of matrix G for the k^{th} node, $k = 1, 2, \ldots, N$. It is easy to see that $K = A\cdot G$ is a symmetric matrix. The pairwise key between the i^{th} node and the j^{th} node is $K_{ji} = K_{ij}$, which can be calculated by the i^{th} node and the j^{th} node after exchanging their columns of G. Blom's scheme is λ-secure, namely the network remains secure if there are no more than λ compromised nodes.

Du *et al.* [15] improved Blom's scheme by giving a new design of matrix G. Instead of storing the whole column of matrix G, a sensor node in Du *et al.*'s scheme only needs to store a key seed which can be used to generate a column of G. A key seed is the second element in each column of matrix G in [15] (as shown in Fig. 2). This helps to reduce the storage cost at the node and the communication cost during key establishment. Another advantage of Du *et al.*'s scheme is the employment of multiple key-space KPS. By viewing the set of keys generated from $A\cdot G$ in Blom's scheme as a key-space, the system authority in Du *et al.*'s scheme generates ω secret symmetric matrices $D_1, D_2, \ldots, D_\omega$ and computes $A_i = (D_i\cdot G)^T$ for each D_i. As a result, there are totally ω key-spaces $A_1\cdot G, A_2\cdot G, \ldots, A_\omega\cdot G$. For the j^{th} sensor node, the system authority randomly selects τ A_is and loads the j^{th} node with the j^{th} row of each selected A_i and the key seed of the j^{th} column of G. Neighbor nodes can successfully establish a pairwise key only if they are loaded with columns from same matrices, which contributes to the improvement of network resilience.

$$G = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ s & s^2 & s^3 & \ldots & s^n \\ s^2 & (s^2)^2 & (s^3)^2 & \ldots & (s^n)^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ s^\lambda & (s^2)^\lambda & (s^3)^\lambda & \ldots & (s^n)^\lambda \end{pmatrix}.$$

FIGURE 2. The public matrix *G* given in Du *et al.*'s scheme [15].

Parakh *et al.* proposed a matrix-based key agreement algorithms for sensor networks in [16]. System authority in their scheme chooses a diagonalizable $N\times N$ symmetric matrix Y at random and diagonalizes Y such that $Y = M^{-1}D_yM$, where D_y is a diagonal matrix with eigenvalues of Y. It then randomly picks a diagonal matrix D_x and computes $N\times N$ matrix $X = M^{-1}D_xM$. It is easy to see that those two matrices commute with each other, i.e., $XY = YX$. Then, system authority randomly picks row-column pairs for each sensor node. For example, system authority randomly picks r from a uniform distribution over $[1, N]$ and assigns node i with the r^{th} row and column of X, and the r^{th} column of Y. After deployment, two sensor nodes i and j can compute two keys K_{ij} and K_{ji} by exchanging their stored columns of Y and agree on a session key $K = Hash(K_{ij} \parallel K_{ji})$. However, scheme in [16] have an inherent flaw: Different links may share the same key due to the random selection of row-column pairs, and thus the security of other links may at risk if a sensor node is captured by adversaries.

C. KEY ESTABLISHMENT SCHEMES EMPLOYING DEPLOYMENT KNOWLEDGE AND/OR MATRIX

Nodes may be deployed following a pre-defined method in certain situations. In nodes deployment using airplane [17], for example, sensors nodes are partitioned into a sequence of groups and dropped out of the airplane sequentially as the airplane flies forward. It is easy to see that sensor groups that are dropped next to each other have a better chance to be close to each other after deployment. By exploiting deployment knowledge in such situations, Du *et al.* [17] extended Eschenauer-Gligor's scheme and proposed a key management scheme. In their scheme, target deployment area is divided into rectangular regions. During key pre-distribution phase, sensor nodes are divided into equal groups $G_{i,j}$ for $i = 1, \ldots, t$ and $j = 1, \ldots, n$. This provides the possibility of setting up multiple key pools with the property that for a key pool $P_{i,j}$, it shares $a\cdot|S_c|$ keys with key pools $P_{i,j-1}$, $P_{i,j+1}$, $P_{i-1,j}$ and $P_{i+1,j}$ ($0 \leq a \leq 0.25$), and $b\cdot|S_c|$ keys with key pools $P_{i-1,j-1}$, $P_{j-1,j+1}$, $P_{i+1,j-1}$ and $P_{i+1,j+1}$ ($0 \leq b \leq 0.25$), where $|S_c|$ is the number of keys each key pool possesses and $4a + 4b = 1$. For each sensor node in

group $G_{i,j}$, system authority selects m keys randomly from its corresponding key pool $P_{i,j}$ and stores those keys into the node. After deployment, neighbor sensor nodes can establish shared keys as in Eschenauer-Gligor's scheme. As claimed in [17], KPS with deployment knowledge can substantially improve a network's connectivity and resilience against node capture with a lower cost of storage.

Du *et al.* further extended the scheme in [17] and proposed a new KPS using deployment knowledge [18]: In key pre-distribution phase, system authority divides N sensor nodes into equal groups $G_{i,j}$ (for $i = 1, \ldots, t$ and $j = 1, \ldots, n$) and generates key-space pools P_{ij}. Employing the similar method in [17] for key-space pools setup, the scheme in [18] achieves the property that two horizontally or vertically neighboring key pools share exactly $a|S_c|$ key-spaces, and two diagonally neighboring key pools share exactly $b|S_c|$ key-spaces, where $0 \leq a \leq 0.25$, $0 \leq b \leq 0.25$ and $4a + 4b = 1$. For each sensor node in group $G_{i,j}$, system authority randomly selects τ key-spaces from its corresponding key-space pool $P_{i,j}$ and stores the corresponding rows of selected matrices A_is into the node. After deployment, neighbor sensor nodes need to find their shared key-space and use the Blom scheme to derive a pairwise key. Their scheme is resilient against node capture with the memory requirement $\tau(\lambda + 1)|q|$ for each sensor node, where key seeds are chosen from GF_q.

Liu *et al.* proposed a key pre-distribution in static sensor networks using both pre-deployment knowledge and post-deployment knowledge [19]. Zhou *et al.* combined the pre-distributed pairwise key scheme with the hash chain, and presented an efficient and scalable pairwise key pre-distribution scheme using deployment knowledge in [20]. Considering that location discovery for sensor networks is very difficult, [21] uses the theory of the signal range and deployment error knowledge to analyze sensor nodes location information, and proved that novel deployment knowledge is also expected to provide superior performance with different types of key distribution schemes.

III. OUR KEY ESTABLISHMENT IN WIRELESS MESH NETWORKS

This section is denoted to the description of our scheme. We will first present preliminaries required in this paper, then followed by a basic scheme (to explain our main idea) and our full scheme.

A. PRELIMINARIES

There are three types of participants in our key establishment scheme, namely system authority, mesh routers and sensor nodes (mesh clients). At a high level, the scheme consists of three phases:

- **System Setup**: System authority generates system parameters;
- **Key Pre-Distribution**: System authority loads each node (including mesh routers and sensor nodes) with pre-loaded key information; and
- **Pairwise Key Establishment**: With the assistance of mesh routers, two sensor nodes establish a secret pairwise key using pre-loaded key information.

We are concerned about:

1) Operations associated with the system authority are carried out in a secure environment, but mesh routers and sensor nodes are not physically secure; In particular, once being captured by adversaries, any sensitive data stored in the nodes will no longer be secret;
2) The area of sensor nodes is within the wireless transmission radius of mesh routers; and
3) Pre-deployment knowledge is available: Most sensor nodes will be deployed to designated regions and only a small number of nodes are deployed to neighbor regions of designated regions, Sensor nodes are static after deployment.

In Table 1 we provide indices of important expressions and equations used throughout this paper.

TABLE 1. Indices.

N	the number of sensor nodes
λ	system's security parameter
s_i	the i^{th} key seed
id_i	identifier of key seed s_i
k_{ij}	the key shared between i^{th} and j^{th} nodes
G	secret matrix
A	public matrix
r	the number of divided subregions
R_i	the i^{th} subregion
β	the number of sensor nodes in each subregion
P_i	the i^{th} sensor group
ε_H	the number of key-space
L^2	the size of target area
t	the length value of regular hexagon
d	the flat-to-flat distance
S_i	the event that the i^{th} key-space is compromised
C_x	the event that x nodes are compromised
B	the event that at least one key-space is broken
P_l	local connectivity
C_c	the computation cost at sensor nodes
Nu_n	the number of shared key-space between neighbor nodes from neighbor regions
Nu_s	the number of shared key-space between neighbor nodes from the same region

B. BASIC SCHEME

Our basic scheme is a variant of Blom's scheme [14] reviewed in Section II.

In Blom's scheme with λ-security, the storage cost at each node is $2 \times (\lambda + 1) \times 128$ bits (if q is a 128-bit number and *AES*-128 is the encryption algorithm). Similarly, each sensor node needs to store $\tau \times (\lambda + 1) \times 128$ bits in the variant of Blom's scheme introduced in [18]. To reduce the storage cost at each sensor node, we modify Blom's scheme as follows.

System Setup: System authority

- Chooses N independent key seeds $s_1, s_2, ..., s_N$ from a finite field GF_q, and let id_i be the identifier of key seed s_i; and

- Generates a *secret* $(\lambda+1)\times N$ matrix G:

$$G=\begin{pmatrix} s_1 & s_2 & \cdots & s_N \\ (s_1)^2 & (s_2)^2 & \cdots & (s_N)^2 \\ \vdots & \vdots & \ddots & \vdots \\ (s_1)^{\lambda+1} & (s_2)^{\lambda+1} & \cdots & (s_N)^{\lambda+1} \end{pmatrix}. \quad (1)$$

Note that in contrast to other matrix-based schemes, G is a **secret matrix** in our scheme.

Key Pre-Distribution: System authority completes the following operations.

Step 1. Operations associated with sensor nodes:

- Stores each key seed s_i and its identifier id_i to the i^{th} sensor node.

Step 2. Operations associated with mesh routers:

- Creates a secret symmetric $(\lambda+1)\times(\lambda+1)$ matrix D in GF_q;
- Computes the public matrix $A=(D\cdot G)^T$; and
- Pre-loads mesh routers with matrix A.

Pairwise Key Establishment:

Step 1. After deployment, each sensor node broadcasts its key seed identifier id_i and keeps a record of all neighbors' key seed identifiers;

Step 2. Mesh routers broadcast matrix A; and

Step 3. Upon receiving matrix A, any two neighbor sensor nodes can establish pairwise keys directly.

Without the loss of generality, let the i^{th} node and the j^{th} node be two neighbors who need to establish a pairwise key. The process is described as follows.

Calculation at the i^{th} node:

- The i^{th} node uses its key seed s_i to calculate the i^{th} column of matrix G: $(s_i, s_i^2, \ldots, s_i^{\lambda+1})^T$;
- Let $(a_{j1}, a_{j2}, \ldots, a_{j(\lambda+1)})$ be the j^{th} row of matrix A, which is broadcast by mesh routers; and
- The i^{th} node calculates the key shared with the j^{th} node as

$$\begin{aligned} k_{ji} &= (a_{j1}, a_{j2}, \ldots, a_{j(\lambda+1)})\cdot(s_i, s_i^2, \ldots, s_i^{\lambda+1})^T \\ &= \sum_{t=1}^{\lambda+1} a_{jt}\cdot(s_i)^t. \end{aligned}$$

Calculation at the j^{th} node:

- The j^{th} node uses its key seed s_j to calculate the j^{th} column of matrix G: $(s_j, s_j^2, \ldots, s_j^{\lambda+1})^T$;
- Let $(a_{i1}, a_{i2}, \ldots, a_{i(\lambda+1)})$ be the i^{th} row of matrix A, which is broadcast by mesh routers; and
- The j^{th} node calculates the key shared with the i^{th} node as

$$\begin{aligned} k_{ij} &= (a_{i1}, a_{i2}, \ldots, a_{i(\lambda+1)})\cdot(s_j, s_j^2, \ldots, s_j^{\lambda+1})^T \\ &= \sum_{t=1}^{\lambda+1} a_{it}\cdot(s_j)^t. \end{aligned}$$

It remains to show that $k_{ij}=k_{ji}$. Note that matrix $K=A\cdot G$ is a symmetric matrix:

$$K=A\cdot G=(D\cdot G)^T\cdot G=G^T\cdot D\cdot G=(A\cdot G)^T. \quad (2)$$

It follows that $k_{ij}=k_{ji}$, i.e., k_{ji} calculated by the i^{th} node is the same as k_{ij} calculated by the j^{th} node.

This completes the description of key establishment between two neighbor nodes.

Remark 1. The major difference between our scheme and others (including Blom's scheme) is the generation of the matrix G. In our scheme, G is a secret matrix and the key seed s_i is given to the i^{th} node, which can only generate the i^{th} column of G but does not have any other information about G. Recall that, as shown in Eq. (1), we choose N independent key seeds and generate secret G matrices. It also explains why we do not generate G matrices as [15], where node i can use its stored key seed s^i to calculate node $2i$'s key seed $s^{2i}=(s^i)^2$ (as shown in Fig. 2). Generating matrix G in this way does not introduce any security issue to other schemes where G is a public matrix but is not applicable in our scheme since G must be a secret.

Remark 2. Another difference between our scheme and others is the involvement of mesh routers during key establishment. Mesh routers in our scheme must broadcast the matrix A, which will be used by sensor nodes to establish pairwise keys. While it also works if system authority pre-loads each sensor node with matrix A, this will introduce additional storage cost at sensor nodes (which are usually resource-constrained devices). In contrast to sensor nodes, mesh routers in WMNs are more powerful and capable of costly operations. By exploiting this heterogeneity, sensor nodes in our scheme can generate pairwise keys in an efficient way.

Remark 3. It is evident that λ must be a large number to provide a certain level of resilience, but a large λ will lead to the increase of computational cost during pairwise key establish phase. To further reduce the computational cost, we employ the deployment knowledge introduced in [17]: target deployment area is divided into multiple regions and most nodes are assumed to be deployed in the pre-defined regions. With this approach, multiple regions correspond to multiple key-spaces which leads to the same level of resilience with a small λ in each key-space. The detailed description is given in Section III-C.

C. OUR FULL SCHEME

Our full scheme is made up of the following phases.

System Setup: System authority

- Divides target deployment area into r regular hexagons, denoted by R_i, $i=1,2,\ldots,r$ (as shown in Fig. 3). It follows that there are around $\beta=\lceil\frac{N}{r}\rceil$ nodes in each region.
- Divides N sensor nodes into r groups denoted by P_i, $i=1,2,\ldots,r$.

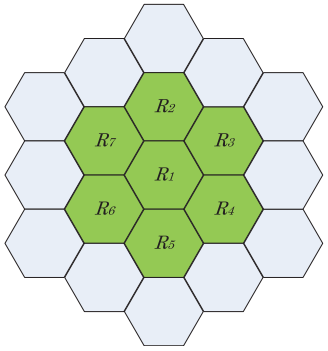


FIGURE 3. Target deployment area is divided into regular hexagons.

- Deployment knowledge: Nodes in group P_i are assumed to be deployed in region R_i.
- Generates N independent key seeds s_{ij}, $i = 1, 2, \ldots, r$ and $j = 1,2,\ldots,\beta$: Let id_{ij} be the identifier of s_{ij}, and key seeds associated with the region R_i are $s_{i1}, s_{i2}, \ldots, s_{i\beta}$.
- Generates $\varepsilon_H = \lceil r-4.32\sqrt{r}+4.6 \rceil$ secret $(\lambda+1)\times(7\cdot\beta)$ matrices G_t for non-edging hexagons, $t = 1, 2, \ldots, \varepsilon_H$: The calculation of ε_H is shown in Section IV.

Each matrix G_t is generated by the key seeds of region R_t and its six neighbor regions in Fig. 3. As an example, G_1 has the form shown in Eq. (3), as shown at the bottom of the page.

Key Pre-Distribution: System authority completes the following operations.

Step 1. Operations associated with sensor nodes:

- Let N_{uw} be the set identifier of G matrices containing key seed s_{uw}; and
- For the w^{th} node in group P_u, load the node with key seed s_{uw}, the corresponding key seed identifier id_{uw} and N_{uw}.

Step 2. Operations associated with mesh routers:

- Generates ε_H secret $(\lambda+1)\times(\lambda+1)$ symmetric matrices D_t and computes public matrices $A_t = (D_t \cdot G_t)^T$, $t = 1, 2, \ldots, \varepsilon_H$; and
- Stores ε_H A matrices in mesh routers.

Pairwise Key Establishment:

Step 1. After deployment, each sensor node broadcasts its key seed identifier id_{uw} and matrix identifier N_{uw}, keeps a record of all neighbors' N_{uw}, and chooses a shared public matrix with each neighbor.

Step 2. Mesh routers broadcast all A matrices; and

Step 3. Upon receiving matrix A, any two neighbor sensor nodes can establish pairwise keys as described prior.

Remark 4. In our scheme, matrix G_t is generated by the key seeds of region R_t and its six neighbor regions. This is due to the fact that while nodes in group P_t are assumed to be deployed in region R_t, it is also likely that some nodes in group P_t may be deployed to the six neighbor regions of R_t. Generating G_t with our approach ensures that any node in group P_t can directly establish pairwise keys with its neighbors if it is deployed to neighbor regions of R_t.

IV. ANALYSIS

This section is devoted to the analysis of our scheme, by comparing it with others [14], [16], [18].

The calculation of ε_H. Let $(L \times L) = L^2\mathfrak{m}^2$ be the size of the target area, divided by r subregions, either regular hexagons or squares (as shown in Fig. 4). It follows that each subregion has the size $A_{re} = \frac{L^2}{r}\mathfrak{m}^2$.

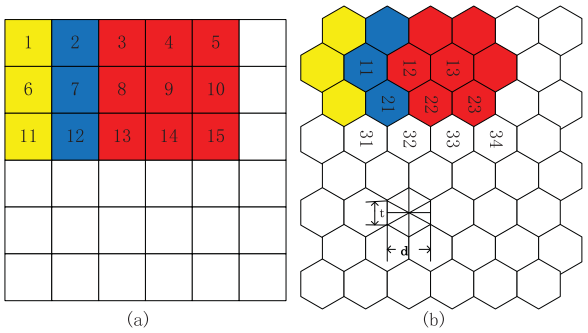


FIGURE 4. Division of target area.

When target area is divided into squares, there are $(\sqrt{r}-2)$ non-edging squares in each row and $(\sqrt{r}-2)$ non-edging squares in each column. Therefore, the number of non-edging areas (i.e., the number of G matrices) is $\varepsilon_S = (\sqrt{r}-2)^2$.

If the same area is divided into regular hexagons, it follows that

$$A_{re} = t \times \frac{\sqrt{3}t}{2} \times \frac{1}{2} \times 6 = \frac{L^2}{r}.$$

Thus, we get

$$t = \frac{\sqrt{2}L}{\sqrt[4]{27}\sqrt{r}}.$$

Compute

$$d = 2 \times \frac{\sqrt{3}t}{2} = \frac{2L}{\sqrt[4]{12}\sqrt{r}},$$

we have

$$\frac{L}{d} = \frac{\sqrt[4]{12}\sqrt{r}}{2}.$$

$$G_1 = \begin{pmatrix} s_{11} & \cdots & s_{1\beta} & s_{21} & \cdots & s_{2\beta} & \cdots & \cdots & s_{71} & \cdots & s_{7\beta} \\ (s_{11})^2 & \cdots & (s_{1\beta})^2 & (s_{21})^2 & \cdots & (s_{2\beta})^2 & \cdots & \cdots & (s_{71})^2 & \cdots & (s_{7\beta})^2 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ (s_{11})^{\lambda+1} & \cdots & (s_{1\beta})^{\lambda+1} & (s_{21})^{\lambda+1} & \cdots & (s_{2\beta})^{\lambda+1} & \cdots & \cdots & (s_{71})^{\lambda+1} & \cdots & (s_{7\beta})^{\lambda+1} \end{pmatrix} \quad (3)$$

Namely, there are $\frac{\sqrt[4]{12}\sqrt{r}}{2}$ regular hexagons in each row. Similarly we can get that there are $\frac{6\sqrt{r}-\sqrt[4]{12}}{3\sqrt[4]{12}}$ regular hexagons in each column. Therefore, the number of non-edging regular hexagons is $\varepsilon_H = (\frac{\sqrt[4]{12}\sqrt{r}}{2} - 2) \times (\frac{6\sqrt{r}-\sqrt[4]{12}}{3\sqrt[4]{12}} - 2) \approx (r - 4.32\sqrt{r} + 4.6) < (\sqrt{r} - 2.16)^2$. Remind that the number of G matrices is $\varepsilon_S = (\sqrt{r} - 2)^2$ when the area is divided into squares. It follows that $\varepsilon_S = (\sqrt{r}-2)^2 > (\sqrt{r}-2.16)^2 > \varepsilon_H$ if $r \geq 4$, i.e., dividing the area into regular hexagons helps to reduce the number of G matrices.

A. SECURITY ANALYSIS

In a hostile environment, an adversary can mount physical attacks on a sensor node after it is deployed and retrieves secret information from its memory. A successful attack on x sensor nodes may affect the security of the network. In this part, we evaluate our full scheme's resilience against node capture attacks. As in [15], our evaluation aims to investigate: the probability that at least one key-space is broken after x nodes are captured. Here we assume that the adversary has no priori knowledge of key seeds stored at nodes so that he compromises nodes randomly.

Probability of at least one key-space being broken. Let S_i be the event that the i^{th} key-space is compromised, for $i \in \{1, ..., \varepsilon_H\}$. We have

$$Pr(B \mid C_x) = Pr(S_1 \cup S_2 \cup ... \cup S_{\varepsilon_H} \mid C_x).$$

Applying the union bound, we get

$$Pr(S_1 \cup S_2 \cup ... \cup S_{\varepsilon_H} \mid C_x) \leq \sum_{i=1}^{\varepsilon_H} Pr(S_i \mid C_x).$$

The fact is that each key-space is broken with equal probability, we have

$$\sum_{i=1}^{\varepsilon_H} Pr(S_i \mid C_x) = \varepsilon_H \cdot Pr(S_1 \mid C_x).$$

Thus, we get,

$$\begin{aligned} Pr(B \mid C_x) &\leq \sum_{i=1}^{\varepsilon_H} Pr(S_i \mid C_x) \\ &= \varepsilon_H \cdot Pr(S_1 \mid C_x). \end{aligned} \tag{4}$$

$Pr(S_1 \mid C_x)$ is the probability that the first key-space is compromised after capturing x nodes. There are ε_H key-spaces in our full scheme, so the probability that a compromised node carries a key seed from the first key-space is $p = \frac{1}{\varepsilon_H}$. Thus, when x nodes are compromised, the probability that j out of these x nodes carry key seeds from the first key-space is $\binom{x}{j} p^j (1-p)^{x-j}$. Recall that a certain key-space can be broken only if at least $\lambda + 1$ nodes are compromised, we get:

$$Pr(S_1 \mid C_x) = \sum_{j=\lambda+1}^{x} \binom{x}{j} p^j (1-p)^{x-j}. \tag{5}$$

Combining Inequality (4) and Eq. (5) , we get the following upper bound:

$$\begin{aligned} Pr(B \mid C_x) &\leq \varepsilon_H \cdot \sum_{j=\lambda+1}^{x} \binom{x}{j} p^j (1-p)^{x-j} \\ &= \varepsilon_H \cdot \sum_{j=\lambda+1}^{x} \binom{x}{j} (\frac{1}{\varepsilon_H})^j (1 - \frac{1}{\varepsilon_H})^{x-j} \end{aligned} \tag{6}$$

We plot analytical results in Fig. 5 where we let $N = 10000$ and $r = 100$. It follows that there are about $\beta = 100$ nodes in each sub-divided region. The figure indicates that: when the system's security parameter is $\lambda = 19$, the adversary have to randomly capture about 100 nodes in order to break at least one key-space with a reasonably-high probability; when λ increases to 29, he need to capture around 150 nodes for the same purpose.

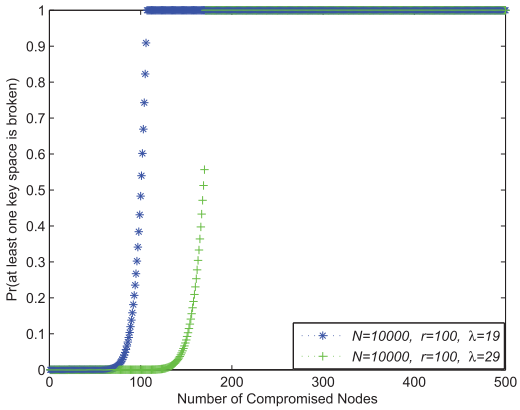


FIGURE 5. Probability of at least one key-space being broken after x nodes are compromised.

Comparison of the number of shared key-spaces. As mentioned above, schemes [14], [16] are single key-space schemes. For the scheme in [18], each sensor node randomly selects τ key-spaces from its corresponding key-space pool. The number of shared key-space is $Nu_{nhv} = (a \cdot P_l \cdot \tau)$ for neighbor sensor nodes from horizontally or vertically regions, $Nu_{nd} = (b \cdot P_l \cdot \tau)$ between neighbor sensor nodes from diagonally regions, and $Nu_s = ((1 - 2a - 2b) \cdot P_l \cdot \tau)$ between neighbor nodes from the same regions at average. For example, let $a = 0.15$, $b = 0.10$, $\tau = \frac{m}{\lambda+1} = \frac{200}{19+1} = 10$, $P_l = 0.3$, then $Nu_n = (Nu_{nhv} + Nu_{nd}) = 0.75$, $Nu_s = 1.5$; let $P_l = 0.5$, then $Nu_n = (Nu_{nhv} + Nu_{nd}) = 1.25$, $Nu_s = 2.5$. In our scheme, the number of shared key-space is $Nu_n = 4$ between neighbor nodes from neighbor regions and $Nu_s = 7$ between neighbor nodes from the same region. Fig. 6 shows their relationship when $a = 0.15$, $b = 0.10$ and $\tau = 10$.

It is easy to see that a large number of key-spaces contributes to a resilient network, i.e., the probability of x captured nodes belong to a unique key-space decreases with the increase of the number of key-spaces. To achieve a high level of security, system authority can compute enough A matrices for each region in key pre-distribution phase. This will lead

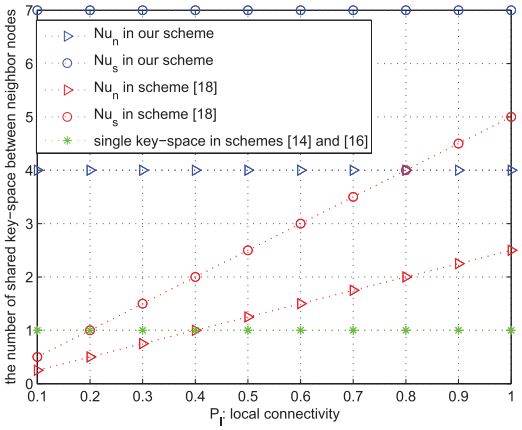


FIGURE 6. The number of shared key-sapce.

to the increase of storage cost at mesh routers, while the consumption at sensor nodes keeps unchanged.

Local connectivity. Local connectivity refers to the probability of any two neighbor nodes sharing at least one key-space [18]. Schemes [14], [16] are single key-space schemes: Key materials stored in sensor nodes are chosen from a single key-space which ensures that any pair of sensor nodes can establish pairwise keys, i.e., local connectivity is 1 in [14], [16]. While scheme [18] makes use of multiple key-space, its local connectivity is affected by a number of parameters, e.g., a, b, τ and $|S_c|$. In our scheme, any pair of neighbor nodes can directly establish a pairwise key, under the assumption that each sensor node is deployed to the designated region or its neighbor regions. So local connectivity in our scheme is 1.

B. OVERHEAD ANALYSIS

Storage complexity. We consider the storage cost of our scheme from two aspects: sensor nodes and mesh routers.

Each sensor node in Du *et al.*'s scheme [18] is associated with τ key-spaces, and for each key-space, the node is loaded with the corresponding row of its matrix A. So, the total number of storage cost is about $\tau \cdot ((\lambda+1)|q| + |q|)$ (Recall that key seeds are chosen from $GF(q)$). Storage costs are $2(\lambda+1)|q|$ and $3N|q|$ in scheme [14] and scheme [16], respectively. In our scheme, each sensor node needs to store a unique key seed ($|q|$), a key seed identifier ($\log_2^N$) and N_{uw} ($\log_2^{\varepsilon_H}$). Therefore, our scheme has a significant advantage over [14], [16], [18] from the aspect of storage cost at sensor nodes.

The light storage cost at the sensor node is achieved by exploiting the heterogeneity of wireless mesh networks: mesh routers have more storage space than sensor nodes. In our scheme, mesh routers need to store ε_H A matrices. The size of each A matrix is determined by the size of G matrix. Again, dividing the area into regular hexagons helps to reduce the size of each G matrix. Recall that each matrix G_t is generated by the key seeds of region R_t and its neighbor regions.

Therefore, the storage cost at mesh routers is $\varepsilon_S \times 9\beta \times (\lambda+1)|q| = 9\beta(\lambda+1)(\sqrt{r}-2)^2|q|$ when deployment area is divided into squares, and the number is $\varepsilon_H \times 7\beta \times (\lambda+1)|q| < 7\beta(\lambda+1)(\sqrt{r}-2.16)^2|q|$ when deployment area is divided into regular hexagons. Here, $\beta = \frac{N}{r}$ is the number of sensor nodes in each group. Fig. 7 shows a specific case when $N = 10000$, and $\lambda = 19$ (we take *AES*-128 as an example). From Fig. 7 we can see that dividing the area into regular hexagons helps to reduce storage cost at mesh routers.

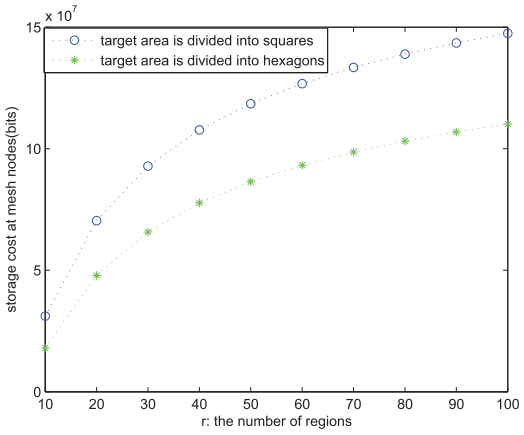


FIGURE 7. Storage cost at mesh routers.

In scheme [14], each sensor node needs to store $2(\lambda+1)|q|$ bits key material, and the total storage cost is $2(\lambda+1)|q| \times N$ bits for the whole networks; Similarly, the total storage cost is $\tau \cdot ((\lambda+1)|q| + |q|) \times N$ in scheme [18], $3N|q| \times N$ in scheme [16] and $\varepsilon_H \times 7\beta \times (\lambda+1)|q| + (|q| + \log_2^N + \log_2^{\varepsilon_H}) \times N$ in our scheme. Fig. 8 shows their relationship when $\lambda = 19$, $\tau = 10$ and $\beta = 100$. Obviously, the total storage cost of our scheme is slightly higher than that of [14], which is lower than that of [16], [18].

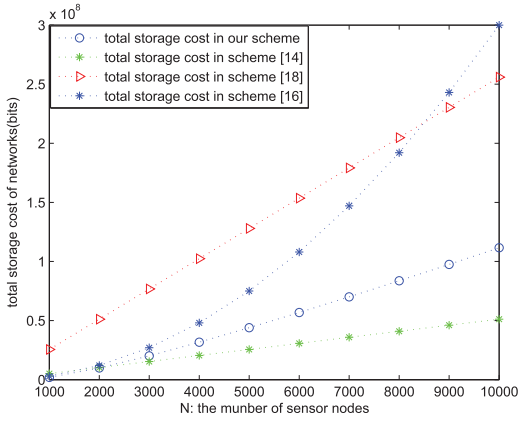


FIGURE 8. Storage cost of networks.

Computation complexity. Schemes [18] need $2\lambda - 1$ multiplication operations in the field $GF(q)$: $\lambda - 1$ multiplications to regenerate a column of G matrix, and λ multiplications to calculate the inner product of the corresponding row-column pairs. Pairwise key establishment between neighbor nodes in

scheme [14] needs about $\lambda + 1$ multiplication operations, in scheme [16] needs about $2N$ multiplication operations. In our scheme, pairwise key establishment between neighbor nodes requires about $2\lambda + 1$ multiplication operations. So, computation complexity is almost the same for our scheme and scheme [18].

We compare the performance of four schemes in Table 2.

TABLE 2. Performance of four schemes.

	Nu_n	Nu_s	P_l	C_c
Our scheme	4	7	=1	$2\lambda + 1$
scheme[14]	-	1	=1	$\lambda + 1$
scheme[18]	$P_l(a+b)\tau$	$P_l(1-2a-2b)\tau$	< 1	$2\lambda - 1$
scheme[16]	-	1	=1	$2N$

Communication complexity. It is analyzed in [11] that communication costs much more than computation during pairwise key establishment. In pairwise key establishment phase, sensor nodes in schemes [14], [16] need to broadcast a column of public matrix; sensor nodes in scheme [18] need to broadcast the indices of selected τ key-spaces, a key seed of public matrix G; and sensor nodes in our scheme only need to broadcast key seed identifier id_{uw} and matrix identifier N_{uw}. It is also pointed out in [18] that local connectivity is one of the dominating factors of communication overhead: If neighbor nodes cannot establish pairwise keys directly, additional operations such as path-key establishment will be necessary which leads to additional communication cost. Comparing with scheme [14], [16], [18], our scheme has a very light communication cost.

C. OTHER ANALYSIS

Key updating. In order to further increase the resilience of networks, we can update pairwise keys when needed.

Firstly, we can update those unexposed key seeds by completing the following operations:

- system authority generates $E_{s_{ij}}(s_{ij}^{'})$ ($s_{ij}^{'}$ is the update of unexposed key seed s_{ij});
- system authority stores $E_{s_{ij}}(s_{ij}^{'})$s in mesh routers; and
- mesh routers broadcast $E_{s_{ij}}(s_{ij}^{'})$s. Obviously, only node with key seed s_{ij} can computes and gains its updated key seed $s_{ij}^{'}$.

As described in our full scheme, pairwise keys can be updated by completing the following operations:

- system authority generates matrices $G_t^{'}$s, $D_t^{'}$s, and computes corresponding matrices $A_t^{'}$s;
- system authority stores matrices $A_t^{'}$s in mesh routers;
- mesh routers broadcast $A_t^{'}$s; and
- sensor nodes update their pairwise keys as described in **Pairwise Key Establishment**.

Scalability. New nodes may be added to the system to replace existing nodes (which are running out of power), or there is a need to sense a new region and extend the network.

To add a new node to replace an existing node in region R_i, system authority

- Selects a new independent key seed $s_{i(\beta+1)}$ and key seed identifier $id_{i(\beta+1)}$ for the new node;
- Updates G matrices associated with R_i and its six neighboring regions;
- Updates corresponding A matrices stored in mesh routers, by choosing new symmetric D matrices;
- Let $N_{i(\beta+1)}$ be the set identifier of G matrices containing key seed $s_{i(\beta+1)}$; and
- Loads the new node with $s_{i(\beta+1)}$, $id_{i(\beta+1)}$ and $N_{i(\beta+1)}$.

After deployment, new added sensor nodes can establish pairwise keys with their neighbors as described in the phase of **Pairwise Key Establishment**.

To discover a new region, we should execute similar operations described in our full scheme.

V. CONCLUSION

Key establishment is a fundamental security issue in wireless mesh networks. This paper presents a new design of matrix-based pairwise key establishment using deployment knowledge in wireless mesh networks. The new scheme has a very light overload of storage and communication at sensor nodes, without introducing any significant computation operations. Furthermore, our scheme is updatable, scalable and secure against node capture attacks. The essential design philosophy of our scheme is the heterogeneity of wireless mesh networks: mesh routers are more powerful than sensor nodes and can afford expensive operations during key establishment. We believe the same idea is also applicable in other situations with the same feature of heterogeneity.

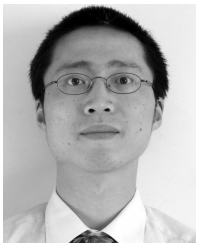
REFERENCES

[1] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Commun. Mag.*, vol. 43, no. 9, pp. S23–S30, Sep. 2005.
[2] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: A survey," *Comput. Netw.*, vol. 47, no. 4, pp. 445–487, 2005.
[3] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. INFOCOM*, 2000, pp. 775–784.
[4] C.-Y. Lin, M.-T. Hung, and W.-H. Huang, "A location-based personal task management application for indoor and outdoor environments," in *Proc. 15th Int. Conf. NBiS*, Sep. 2012, pp. 582–587.
[5] A. P. Lauf, R. A. Peters, and W. H. Robinson, "A distributed intrusion detection system for resource-constrained devices in ad-hoc networks," *Ad Hoc Netw.*, vol. 8, no. 3, pp. 253–266, 2010.
[6] C. Pham, "Scheduling randomly-deployed heterogeneous video sensor nodes for reduced intrusion detection time," in *ICDCN* (Lecture Notes Comput. Sci.), vol. 6522. Berlin, Germany: Springer, 2011, pp. 303–314.
[7] W. Galuba, P. Papadimitratos, M. Poturalski, K. Aberer, Z. Despotovic, and W. Kellerer, "Castor: Scalable secure routing for ad hoc networks," in *Proc. IEEE INFOCOM*. Mar. 2010, pp. 2829–2837.
[8] S. Khan, N. A. Alrajeh, and K.-K. Loo, "Secure route selection in wireless mesh networks," *Comput. Netw.*, vol. 56, no. 2, pp. 491–503, 2012.
[9] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proc. 3rd IEEE Int. Conf. PerCom*, Mar. 2005, pp. 324–328.
[10] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," *IEEE Commun. Mag.*, vol. 32, no. 9, pp. 33–38, Sep. 1994.
[11] D. W. Carman, P. S. Kruus, and B. J. Matt, "Constraints and approaches for distributed sensor network security (final)," DARPA Project Rpt., Cryptographic Technol. Group, Trusted Inform. Syst., NAI Labs, Tech. Rep. 00-010, 2000.

[12] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. ACM Conf. Comput. Commun. Security*, 2002, pp. 41–47.
[13] H. Chan, A. Perrig, and D. X. Song, "Random key predistribution schemes for sensor networks," in *Proc. IEEE Symp. Security Privacy*, May 2003, pp. 197–213.
[14] R. Blom, "An optimal class of symmetric key generation systems," in *Advances in Cryptology* (Lecture Notes in Comput. Sci.), vol. 209, T. Beth, N. Cot, and I. Ingemarsson, Eds. Berlin, Germany: Springer-Verlag, 1985, pp. 335–338.
[15] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Trans. Inf. Syst. Security*, vol. 8, no. 2, pp. 228–258, 2005.
[16] A. Parakh and S. Kak, "Matrix based key agreement algorithms for sensor networks," in *Proc. IEEE 5th Int. Conf. ANTS*, Dec. 2011, pp. 1–3.
[17] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proc. INFOCOM*, Mar. 2004, pp. 586–597.
[18] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," *IEEE Trans. Dependable Secure Comput.*, vol. 3, no. 1, pp. 62–77, Jan./Mar. 2006.
[19] D. Liu and P. Ning, "Improving key predistribution with deployment knowledge in static sensor networks," *ACM Trans. Sensor Netw.*, vol. 1, no. 2, pp. 204–239, 2005.
[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," *Comput. Commun.*, vol. 32, no. 1, pp. 124–133, 2009.
[21] N. T. T. Huyen, M. Jo, T. D. Nguyen, and E. Nam Huh, "A beneficial analysis of deployment knowledge for key distribution in wireless sensor networks," *Security Commun. Netw.*, vol. 5, no. 5, pp. 485–495, 2012.
[22] K. Ren and W. Lou, "A sophisticated privacy-enhanced yet accountable security framework for metropolitan wireless mesh networks," in *Proc. 28th ICDCS*, 2008, pp. 286–294.
[23] K. Ren, K. Zeng, and W. Lou, "A new approach for random key predistribution in large-scale wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 6, pp. 307–318, 2006.

YUEXIN ZHANG received the B.S. degree from the Department of Physics and Electronic Information Engineering, Inner Mongolia Normal University, China, in 2010, and the M.S. degree from the School of Mathematics and Computer Science, Fujian Normal University, China, in 2013. His research interests include network security.

LI XU is a Professor and Doctoral Supervisor at the School of Mathematics and Computer Science, Fujian Normal University. He received the B.S. and M.S. degrees from Fujian Normal University in 1992 and 2001, respectively, and the Ph.D. degree from the Nanjing University of Posts and Telecommunications in 2004. He is currently the Vice Dean of the School of Mathematics and Computer Science and the Director of the Key Laboratory of Network Security and Cryptography, Fujian. His interests include wireless networks and communication, network and information security, complex networks and systems, and intelligent information in communication networks. He has been invited to act as PC chair or member at more than 30 international conferences. He is a member of IEEE and ACM, and a senior member of CCF and CIE in China. He has published over 100 papers in refereed journals and conferences.



YANG XIANG received the Ph.D. degree in computer science from Deakin University, Australia. He is currently a Full Professor with the School of Information Technology, Deakin University. He is the Director of the Network Security and Computing Laboratory. His research interests include network and system security, distributed systems, and networking. He is currently leading his team developing active defense systems against large-scale distributed network attacks. He is the Chief Investigator of several projects in network and system security, funded by the Australian Research Council. He has published more than 130 research papers in many international journals and conferences, such as the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON INFORMATION SECURITY AND FORENSICS, and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. Two of his papers were selected as the featured articles in the April 2009 and the July 2013 issues of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. He has published two books *Software Similarity and Classification* (Springer) and *Dynamic and Advanced Data Mining for Progressing Technological Development* (IGIGlobal). He has served as the Program/General Chair for many international conferences such as ICA3PP 12/11, IEEE/IFIP EUC 11, IEEE TrustCom 13/11, IEEE HPCC 10/09, IEEE ICPADS 08, NSS 11/10/09/08/07. He has been the PC member for more than 60 international conferences in distributed systems, networking, and security. He serves as an Associate Editor of the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, *Security and Communication Networks* (Wiley), and the Editor of the *Journal of Network and Computer Applications*. He is the Coordinator, Asia for the IEEE Computer Society Technical Committee on Distributed Processing.

XINYI HUANG received the Ph.D. degree from the School of Computer Science and Software Engineering, University of Wollongong, Australia, in 2009. He is currently a Professor with the Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, China. His research interests include cryptography and information security. He has published over 60 research papers in refereed international conferences and journals. His work has been cited more than 1000 times at Google Scholar. He is in the Editorial Board of the *International Journal of Information Security* (IJIS, Springer) and has served as the Program/General Chair or Program Committee Member in over 40 international conferences.