

# Secure Pairwise Key Establishment in Large-scale Sensor Networks: An Area Partitioning and Multi-group Key Predistribution Approach

Dijiang Huang

Arizona State University

Deep Medhi

University of Missouri–Kansas City

---

Existing pairwise key establishment schemes for large-scale sensor networks are vulnerable to various passive or active attacks. We classify attacks as: selective node capture attacks, node fabrication attacks, and insider attacks. In order to improve the security robustness of random key predistribution and pairwise key establishment schemes against these attacks, we propose a 5-phase pairwise key predistribution and pairwise key establishment approach by using area partitioning and multi-group key predistribution. Our security performance studies show that our proposed approach is resilient to selective node capture and node fabrication attacks, and restricts the consequence of any insider attack to a minimal level.

Categories and Subject Descriptors: C.2.0 [**COMPUTER-COMMUNICATION NETWORKS**]: General—*Security and protection*; C.2.4 [**COMPUTER-COMMUNICATION NETWORKS**]: Distributed Systems —*Distributed applications*

General Terms: Security

Additional Key Words and Phrases: Sensor, Selective Node Capture, Node Fabrication, Insider attack

---

## 1. INTRODUCTION

Large-scale distributed sensor networks are composed of a large number of low-power sensor devices, for example, SmartDust[Kahn et al. 1999] and WINS [Pottie and Kaiser 2000]. Typically, these networks are installed to collect sensed data from sensors deployed in a large area, as shown in Figure 1. Sensor networks often have one or more centralized controllers called base stations or sinks. A base sta-

---

© ACM, 2007. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in PUBLICATION, VOL , NO , August 2007 <http://portal.acm.org/browse.dl.cfm?idx=J789>

Authors address: Dijiang Huang, Computer Science and Engineering Department, Arizona State University, 699 S. Mill Ave. Suite 470, Tempe, AZ. 85287-8809. Deep Medhi, Computer Science and Electrical Engineering Department, University of Missouri, 550F Flarsheim Hall, 5100 Rockhill Road, Kansas City, MO. 64110; email:dijiang@asu.edu, dmedhi@umkc.edu.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 2007 ACM 0000-0000/2007/0000-0001 \$5.00

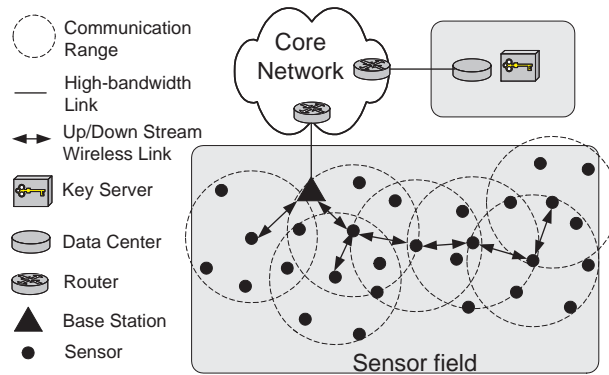


Fig. 1. Sensor network architecture.

tion, usually many orders of magnitude more powerful than a sensor, is typically a gateway to other networks or data centers via high bandwidth communication links (either leased lines or broadband wireless links). They can be used as a nexus to disseminate control information into the network or extract data from it. Sensors are constrained to lower-power and lower-bandwidth usage; thus, most deployed sensors may not be able to communicate with the base station directly requiring sensed data to be aggregated at some forwarding nodes. As a result, communication types should be either one-to-many to disseminate control commands from the base station to each sensor, or many-to-one to collect sensed data from each sensor to the base station. For example, as shown in Figure 1, double-head lines form up/down stream wireless links which connect all aggregation nodes. Aggregation nodes can be dynamically selected, for example, using an election protocol (e.g., LEACH [Ganesan et al. 2001], TEEN [Manjeshwar and Agrawal 2001], and PEGASIS [Lindsey and Raghavendra 2002]) where each aggregation node collects sensor readings from surrounding nodes and forwards a single message containing aggregated information. In this way, a multi-hop wireless network is formed to allow sensors to communicate to the nearest base station. Thus, in large-scale distributed sensor networks, a critical security service is to provide direct secure communication channels or links among sensors.

As specified in [Akyildiz et al. 2002], the number of sensor nodes deployed for studying a phenomenon may be on the order of hundreds of thousands. Depending on the application, the number may reach an extreme value of millions. Due to inherent storage constraints, it is infeasible for a sensor device to store a unique shared key for every other sensor in the system. A naïve solution is to use a common key between every pair of sensors which can overcome storage constraints, but offers weak security. In this case, if just a single node is compromised, the entire system is compromised. Recently, Random Key Predistribution (RKP) schemes were proposed [Eschenauer and Gligor 2002; Chan et al. 2003; Liu and Ning 2003a; Liu et al. 2005; Du et al. 2003; Du et al. 2005] for large-scale distributed sensor networks. These schemes randomly select a subset of keys from a large key pool

for each sensor. Since the RKP schemes preinstall a limited number of keys in each sensor, after being deployed, a sensor is not guaranteed to share a key with each of its neighbors. Thus, a sensor can establish pairwise keys via those neighbors with which it already shares preinstalled keying materials. For RKP, a Pairwise Key Establishment (PKE) protocol is needed to set up shared keys with neighbors.

### 1.1 Issues of Existing RKP Schemes

For current RKP schemes, analyses of the security strength are done on the basis of the number of communication links that can be compromised due to compromised sensors in the network; furthermore, compromised nodes are randomly captured in existing security analysis models. To mitigate the random node capture attack, Du et al. [Du et al. 2004], Liu and Ning [Liu and Ning 2003b; 2005] proposed using deployment information (sensor location information) to improve the resilience to the node capture attack. However, in practice, the open or hostile deployment environment of sensor networks makes it easier for attackers to locate and selectively capture sensors. Moreover, due to the lack of node authentication, attackers can easily fabricate nodes by using the secrets which are preinstalled in the captured nodes.

It may be noted that PKE protocols are vulnerable to insider attacks, in which the attacker can fabricate captured sensors, implant malicious codes, and deploy the fabricated sensors back into the sensor networks. These fabricated sensors can malfunction the PKE protocol by sniffing the encrypted key message, dropping, forging, and redirecting the key message, etc.

### 1.2 Overview of Proposed Approach

In this paper, we propose a comprehensive solution for the RKP-PKE scheme to prevent and mitigate various types of attacks. We present our scheme in five phases. In our scheme, a sensor deployment area is first partitioned into multiple small square areas (zones) and then, sensors deployed in each zone form a group. This design can restrict the consequence of attacks (such as insider attacks) within a small range. We utilize the unconditionally secure and  $\lambda$ -collusion resistant properties of the group keying scheme proposed in [Blundo et al. 1998] to construct key spaces and restrict the number of deployed secrets of a key space to no more than  $\lambda$ . In this way, we can effectively prevent attackers from sniffing traffic and fabricating new sensors via captured keys. To improve resilience to insider attacks, we propose a source routing based multi-path PKE protocol. This multi-path PKE protocol utilizes  $(n, k)$  Reed-Solomon error correcting codes to set up pairwise keys and it is resilient to  $t = (n - k)/2$  paths are faulty.

### 1.3 Main Contributions

Our main contributions are in two directions:

- (1) We propose a  $\lambda$ -restricted area partitioning and multi-group key predistribution scheme to scale the network size due to the key storage constraints of sensors. Compared to previous work, our scheme can completely prevent selective node capture attacks. In addition, attackers cannot fabricate new sensors to set up pairwise keys with uncompromised sensors in our scheme. Our scheme also

restricts the node replication attack within the partitioned zone where the node is captured. Thus, it is more difficult for attackers to duplicate captured sensors and to distribute them in other zones in order to compromise the entire sensor network.

- (2) We use the multiple node-disjoint-path pairwise key establishment protocol and fault-tolerance coding scheme (Reed-Solomon codes) to improve resilience to insider attacks, such as stop forwarding and cheating. With our scheme, a sensor would be able to recover the key establishment message due to packet dropping as well as to identify faulty paths that alter a forwarded key message.

#### 1.4 Organization

The rest of the paper is organized as follows: In Section 2, we present the related work. In section 3, we highlight the proposed five phases of our pairwise key setup scheme and attack model. The details of our scheme is presented in Section 4. From Section 5 to Section 7, we discuss the performance of our scheme against selective node capture attacks, node fabrication attacks, and insider attacks. The operational performance assessments such as communication, storage, and computation, are presented in Section 8. In Section 9, we present a summarization of our approach and the security challenges for RKP-PKE schemes.

## 2. RELATED WORK

We broadly classify the random key predistribution (RKP) schemes for sensor networks into two groups: Purely Random Key Predistribution (P-RKP) schemes and Structured Key-pool Random Key Predistribution (SK-RKP) schemes. The P-RKP scheme was first proposed by Eschenauer and Gligor [Eschenauer and Gligor 2002], and we refer to it as the *basic scheme*. Several schemes have been developed which were all based on the *basic scheme*; these schemes improve the *basic scheme* in five aspects: 1) shared keys threshold:  $q$ -composite scheme [Chan et al. 2003]; 2) key pool structure: SK-RKP schemes [Liu and Ning 2003a; Liu et al. 2005; Du et al. 2003; Du et al. 2005]; 3) path-key establishment protocol:  $k$ -path key establishment schemes [Chan et al. 2003; Zhu et al. 2003]; 4) location awareness schemes [Liu and Ning 2003b; 2005; Du et al. 2004]: key predistribution and sensors' deployment are based on known sensors deployment information; 5) shared keys discovery schemes [Di Pietro et al. 2004; Mehta et al. 2005]: one-way function schemes have been proposed to reduce the communication overhead during the key discovery phase in order to improve resilience to a node fabrication attack.

Note that the above discussed schemes assume that the attacker randomly captures sensors in order to compromise the sensor network. However, we argue that in reality, attackers are often very smart and they might be able to figure out attacking strategies to maximize gains with minimal attacking efforts. To analyze more sophisticated attacks, a preliminary analysis of selective node capture attack and node fabrication attack was presented in [Huang et al. 2004].

Before we discuss additional related work, we state a few terms from existing literature: if two neighbors share a preinstalled key, the key is called a *direct key*. If two neighbors do not share a preinstalled key, they need to find a path that is protected by *direct keys* to establish a pairwise key. The established pairwise key

is called an *indirect key*. To safeguard the *indirect key*, multiple key-path schemes have been proposed in [Chan et al. 2003] and [Zhu et al. 2003] to prevent faulty sensors from deriving *indirect keys*. In [Chan et al. 2003], multiple *physically* link-disjoint paths between two nodes are used to set up an *indirect key*. When two nodes  $u$  and  $v$  want to set up an *indirect key* via multiple (say  $j > 1$ ) link-disjoint paths, the source node,  $u$ , selects  $j$  secrets,  $s_1, \dots, s_j$ , and sends each of these secrets onto a unique key establishment path. To secure a secret message,  $s_1$ , via a key establishment path,  $u \rightarrow x \rightarrow v$ , following key establishment steps are performed:

$$u \rightarrow x : \{s_1\}_{k_{ux}}; x \rightarrow v : \{s_1\}_{k_{xv}},$$

where  $k_{ux}$  and  $k_{xv}$  are *direct keys* shared between pair  $(u, x)$  and pair  $(x, v)$ , respectively. Upon receiving all the secrets, node  $v$  simply uses bitwise *XOR* operation to derive the *indirect key*, i.e.,

$$\textit{indirect\_key} = s_1 \oplus \dots \oplus s_j. \quad (1)$$

In [Zhu et al. 2003], multiple *logical* link-disjoint paths between two nodes are used for setting up an *indirect key*. A *logical* path means that there exists a key sharing relation among source, destination, and intermediate nodes along the key establishment path. For example, source node  $u$  shares  $t_1$  *direct keys* with intermediate node  $x$ , and node  $x$  shares  $t_2$  *direct keys* with destination node  $v$ ; note that  $u$  and  $v$  do not share a *direct key*. Since a *direct key* can be only used for one *logical* path, there can be  $z_x = \min(t_1, t_2)$  key establishment paths between  $u$  and  $v$  via intermediate node  $x$ . The secrets selection and transmission proposed in [Zhu et al. 2003] is similar to that described in [Chan et al. 2003]. The difference is the use of *physical* or *logical* key establishment paths in corresponding proposed schemes.

Both proposed multi-path key establishment schemes are efficient to guard against outsider's node capture attacks and insider attacks that passively learn the forwarded messages. However, they are vulnerable to active insider attacks, i.e., an attacker can stop forwarding secrets or alter the forwarded secrets which can prevent the receiver from deriving the right *indirect key*. In order to counter active insider attacks, we have earlier proposed a Reed-Solomon code and multi-path key establishment protocol to enable a sensor to identify the faulty key establishment paths in a preliminary study [Huang and Medhi 2005].

The local key-graph connectivity is important for evaluating the communication overhead and storage overhead of a RKP-PKE approach. Random graph theory [Spencer 2001] based approaches have been adopted by existing RKP-PKE solutions such as [Eschenauer and Gligor 2002]. In a recent work [?], it has been shown that the random graph based solutions introduce errors when the group size is either big or small. Moreover, the random graph based solution is unable to provide the key-path length information. These problems are solved by using a modified binomial distribution in a hop-by-hop fashion to evaluate the local key graph connectivity for a key path within  $h$  hops. However, multipath key graph connectivity (within 2 hops) and predistribution approaches were not addressed in [?]  
—the scope of this paper is to address these aspects along with a comprehensive analysis of node fabrication and node capture attacks.

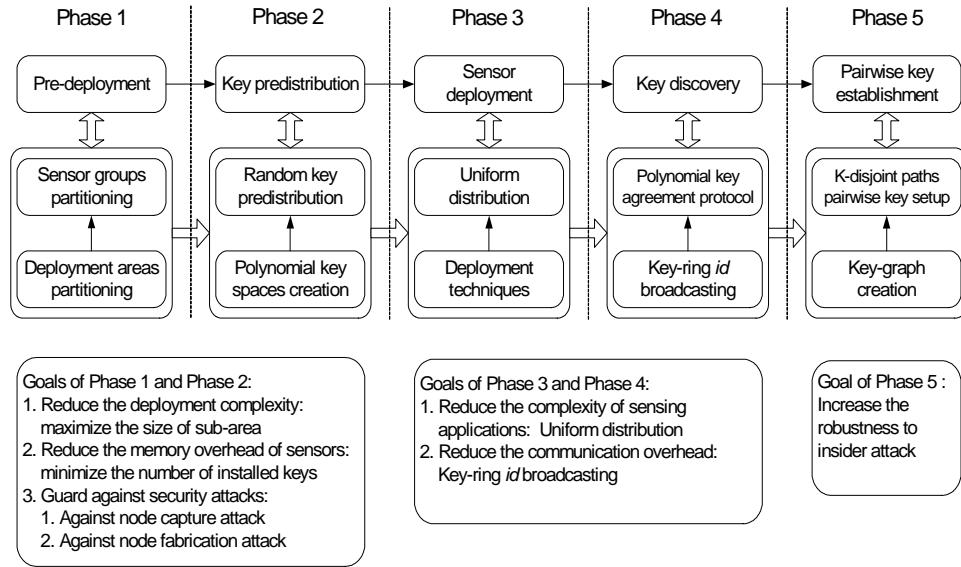


Fig. 2. A taxonomy of RKP-PKE schemes.

### 3. SYSTEM MODELS OF OUR SCHEMES

#### 3.1 A Generalized RKP-PKE Model

Our approach involves five phases as shown in Figure 2. Here, we highlight the functionalities for each phase of these five phases:

1. *Pre-deployment phase:* A sensor deployment area is first partitioned into multiple sub-areas. A group of sensors is predetermined for deployment in a sub-area.
2. *Key predistribution phase:* A centralized key server generates a structure key pool for each sensor group. A structure key pool is composed of multiple key spaces (key matrices). The key predistribution is described later in Section 4.2. After key predistribution, each sensor is assigned a unique key ring *id*. The key ring *id* can identify the sensor's position in each key space, and thus, it helps to establish pairwise keys with other sensors in the key discovery phase and the pairwise key establishment phase.
3. *Sensor deployment phase:* Sensors are deployed in a two-dimensional plane. Based on group partitions, each group of sensors are uniformly<sup>1</sup> deployed in each sub-area.
4. *Key discovery phase:* Once sensors are deployed, three steps are involved in the key discovery phase.
  - step 1:* Each sensor broadcasts its key-ring *id* to all its neighbors.
  - step 2:* Based on the received key-ring *id*, a sensor can determine its neighbor-reachability list, and then it broadcasts its neighbor-reachability list.

<sup>1</sup>For discussion on why we emphasize on using uniform distribution, see Section 4.3.

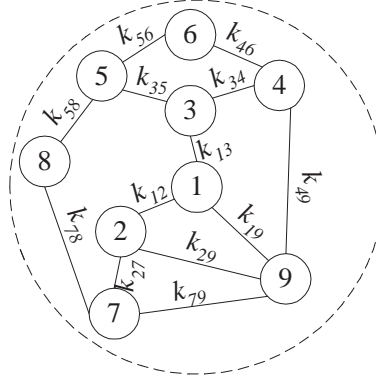


Fig. 3. A local key graph for sensor 1.  $k_{xy}$  represents a direct key between  $x$  and  $y$  and the dot circle represents the communication range of sensor 1. In this key graph, node 1 can set up indirect keys with all its communication neighbors within 3 hops.

*step 3:* Based on the received neighbor-reachability list, a sensor can build a *local key graph* (shown in Figure 3) which represents the node-connectivity from the view of each individual node. We note that in the *local key graph*, if two sensors share a *direct key*, we refer to it as a link connecting two sensors directly.

5. *Pairwise key establishment phase:* In a *local key graph*, if there is no link between two sensors, i.e., they do not share a *direct key*, the PKE protocol will be used to set up an *indirect key* via multiple hops (i.e., a key path) in the *local key graph*. The established pairwise key is called an *indirect key*. The *indirect key establishment* includes two phases:
  - (a) The key establishment within a zone.
  - (b) The key establishment among adjacent zones.

In order to counter insider attacks, such as the insider broadcast incorrect neighbor-reachability list and drop and/or alter the key establishment packets, a multi-path key establishment protocol is proposed to counter insider attacks.

### 3.2 Attack Model

As classified in [Karlof and Wagner 2003], a laptop-class attacker can have more powerful devices, like laptops or their equivalents. Thus, they have an advantage over legitimate nodes: they may have greater battery power, a more capable CPU, a high-power radio transmitter, or a sensitive antenna. We assume that the goal of attackers is to compromise the pairwise keys established or to be established between any two sensors. Furthermore, the following capabilities of an attacker are assumed:

- The attacker has unlimited energy and computing power.
- The attacker knows all the information stored in a sensor once the sensor is captured.
- The attacker can listen to and record all the traffic in the network.

- The attacker has the ability to physically locate a given sensor by listening to the traffic.
- The attacker has the ability to fabricate similar nodes, deploy, and control them.

We classify attacks to RKP-PKE schemes into three categories.

- Selective node capture attack: attacking communication link*<sup>2</sup>. Most of previous security analysis for RKP-PKE schemes assumes that attackers randomly capture sensors to compromise preinstalled keys. However, the random node-capturing assumption is weak. For example, an attacker can purposefully locate active sensors and compromise the sensors which can give him more information of the sensor networks; i.e., the attacker wants to capture the minimal number of sensors to compromise the maximum number of preinstalled keys. To evaluate the resilience to selective node capture attack of a RKP-PKE scheme, we measure its security performance by computing the fraction of the number of compromised network links by capturing  $x$  nodes, where the compromised network links do not include the links connect to  $x$  compromised nodes.
- Node fabrication attack: attacking authenticity*. The attacker can insert new nodes into the network after obtaining some secret information. This is a serious attack since the compromise of even a single node might allow an attacker to populate the network with clones of the captured node to such an extent that legitimate nodes could be outnumbered and the attacker can thus gain full control of the network. We evaluate the resilience toward node fabrication by estimating the fraction of total uncompromised sensor nodes that can set up communication links with the fabricated nodes by capturing  $x$  nodes.
- Insider attack: attacking PKE Protocol*. Since the *indirect keys* are established via multiple *key paths* set up during the key discovery phase. the attacker can capture the *indirect keys* by helping uncompromised nodes to set up pairwise keys. Before the attacker deploys this attack, it itself must be a valid node (an insider) in the system (e.g., via node fabrication attack). We evaluate this attack by the probability that an *indirect key* is compromised due to  $x$  compromised nodes. Resilience to insider attacks can be achieved by using our proposed multiple node-disjoint-path PKE scheme in Section 7.

Note that existing RKP-PKE approaches are all vulnerable to the above attacks.

#### 4. USING AREA PARTITIONING, MULTI-GROUP KEY PREDISTRIBUTION, AND MULTI-PATH PKE

In this section, we elaborate our 5-phase pairwise key establishment scheme. The proposed area partitioning, group partitioning, and SK-RKP scheme will be able to determine the size of a zone (sub-area) and the maximum number of sensors deployed within a zone. The security goals of our approach are to counter any selective node capture attacks, node fabrication attacks, and insider attacks. The notations used in the following sections are given in Table I.

<sup>2</sup>We consider a communication link as a direct communication channel between two neighboring sensors which protected by a pairwise key. Once the pairwise key is compromised, the corresponding communication link is compromised.

Table I. Notations.

Notations for sensor deployment	
$a$	Size of a zone
$r$	Sensor communication radius
$(i, j)$	Zone indexed by $i$ and $j$
$Z(i, j)$	Zone $(i, j)$
$\mathcal{N}_b(i, j)$	Number of neighbors of sensor $b$ deployed in $Z(i, j)$
$n_z$	the number of nodes in a group
$u, v$	Sensor $u$ and $v$
$[(i, j), u]$	Sensor $u$ 's $id$
$n$	Number of neighbors of a sensor
$N$	Total number of deployed sensors
Notations for probabilities of key predistribution	
$\mathcal{P}$	A large key pool
$\mathcal{P}(i, j)$	Key spaces associated with group $G(i, j)$
$P$	Size of the key pool, $P =  \mathcal{P} $
$m$	Number of keys preinstalled in a sensor
$\lambda$	Distribution threshold of a key space
$\omega$	Number of key spaces
$\tau$	Number of sub-key spaces selected for each sensor
$D$	A secret matrix size of $(\lambda + 1) \times (\lambda + 1)$
$G$	A publicly know matrix size of $(\lambda + 1) \times N$
$A$	A secret matrix = $(D \cdot G)^T$
$K$	A key matrix = $A \cdot G$
$p_1$	Probability that two sensors share a preinstalled key
Notations for attacks	
$C_x$	Number of compromised keys by compromising $x$ nodes
$x$	Number of compromised nodes
$R(x)$	Fraction of compromised links among uncompromised nodes

#### 4.1 Phase 1: Area Partitioning and Group Partitioning

We assume that the sensor deployment area is a two-dimensional rectangular region with the size  $(i \cdot a)(j \cdot a)$  square meters. The rectangular region can be further divided into  $(i \times j)$  deployment areas, each of size  $a^2$  square meters. In this paper, we denote each small deployment area as a *zone*  $Z(i, j)$ . An example of a deployment region is shown in Figure 4, where  $i = j = 6$ .

We use  $G(i, j)$  to denote the group of sensors deployed in the zone  $Z(i, j)$ . We assume that the sensors are uniformly distributed over the deployment area for each group, and the number of sensors in a group is  $n_z$ . We denote the total number of sensors in the entire deployment region by  $N$ . Thus, we have  $N = n_z \cdot i \cdot j$ . A sensor is identified by  $[(i, j), u]$ , where  $(i, j)$  is the group  $id$ , and  $u$  is the unique node  $id$ , where  $u = 1, \dots, N$ . We use  $\rho = n_z/a^2$ , where  $\rho$  is a constant under the assumption of uniform distribution, to represent the density of the sensor deployment within a zone. If the density  $\rho$  is given (based on the manufacture's information), the size of a zone must fulfil the inequality

$$a^2 \leq n_z/\rho. \quad (2)$$

In phase 2, we will use the SK-RKP scheme for the key predistribution scheme within a zone (the detailed description of SK-RKP scheme is given in Appendix A).

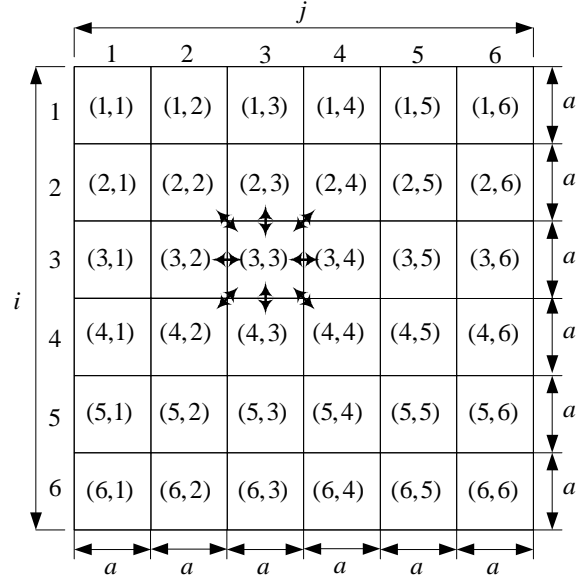


Fig. 4. Sensor deployment in a grid structure.

To achieve the *non-colluding*<sup>3</sup> property, we restrict that no more than  $\lambda$  sensors are allowed to choose a given key space, where  $\lambda$  is a parameter of the SK-RKP scheme and  $\lambda$  restricts the size of a key matrix.

According to Blundo et al. [Blundo et al. 1998], for each key space, the key matrix  $A$  is a  $N \times (\lambda + 1)$  matrix. If an attacker has the knowledge of more than  $\lambda$  rows, the entire matrix  $A$  can be derived. Thus, we restrict the number of rows distributed to sensors for each key matrix  $A$  to be no more than  $\lambda$ . Under this restriction, we have the relation

$$n_z \leq \lambda\omega/\tau, \quad (3)$$

where  $\omega$  is the total number of key matrix within a key pool, and  $\tau$  is the number of key matrices selected by a sensor. Once  $\omega$  and  $\tau$  are determined, the zone size increases linearly with the increase of  $\lambda$ . From (2) and (3), we can derive the relation between zone size  $a^2$  and key matrix parameters:

$$a^2 \leq \lambda\omega/(\tau\rho). \quad (4)$$

#### 4.2 Phase 2: Key Predistribution

The key predistribution scheme used within a group is called *I-Scheme* and the key predistribution scheme used between two neighboring groups is called *E-Scheme*. The *I-Scheme* is presented as follows:

<sup>3</sup>The *non-colluding* feature of the proposed pairwise key scheme is described as follows: for all pairwise key  $k_{uv}$ , where  $k_{uv}$  is the pairwise key that can be derived from the secrets possessed by sensors  $u, v \in \mathcal{S}$ , where  $\mathcal{S}$  is the set of all sensors deployed in the system, all sensors in the set  $\mathcal{S} \setminus \{u, v\}$  cannot derive the pairwise key  $k_{uv}$ .

#### 4.2.1 *I-Scheme*

- i. Key pool  $\mathcal{P}$  is composed of multiple sub-key pools. Each sub-key pool is associated with a small partitioned area and a sub-key pool is represented as  $\mathcal{P}(i, j)$ . By using the SK-RKP scheme, each sub-key pool is divided into  $\omega$  key spaces. A key space is a  $N \times (\lambda + 1)$  key matrix  $A$ . Each element of  $A$  is a unique key.
- ii. Divide the  $N$  sensors in groups. According to the partitioned area, a group is represented by  $G(i, j)$ .
- iii. Assign unique identifiers to the sensors. For each sensor, assign the  $id = [(i, j), u]$ , where  $(i, j)$  is the group  $id$ , and  $u = 1, \dots, N$ .
- iv. For sensor  $[(i, j), u]$ , randomly select  $\tau$  key spaces from  $\omega$  key spaces in  $\mathcal{P}(i, j)$  while making sure that the selected sub-key space is not already selected  $\lambda$  times; load the sensor with the  $u^{th}$  row of matrix  $A$  for each selected sub-key space.

For details about the SK-RKP scheme, refer to Appendix A.

4.2.2 *E-Scheme*. For *E-Scheme*, as shown in Figure 4, a zone can have the maximum of 8 neighboring zones; e.g., the bidirectional arrows are shown around zone  $Z(3, 3)$ . The key predistribution scheme (*E-Scheme*) for sensors in two adjacent zones is given as follows:

- i. For a sensor  $u$  in group  $G(i_1, j_1)$ , randomly select one sensor, say  $v$ , from one of its neighboring groups, say  $G(i_2, j_2)$ . Groups  $G(i_1, j_1)$  and  $G(i_2, j_2)$  are neighbors if  $|i_1 - i_2| \leq 1$  or  $|j_1 - j_2| \leq 1$ .
- ii. Install the duple,  $(k_{uv}, id_v)$ , in  $u$  and the duple,  $(k_{uv}, id_u)$ , in  $v$ , where key  $k_{uv}$  is unique and  $id_u, id_v$  are the identifiers of node  $u, v$ , respectively. Once node  $u$  selects a peer node  $v$  in group  $G(i_2, j_2)$ , it cannot select another node in the same group.
- iii. If all sensors have selected a node in each of its neighboring groups, stop; otherwise goto i.

Using *E-scheme*, each sensor maintains a unique key shared with a single node in each of its neighboring zones.

### 4.3 Phase 3: Sensor Deployment

We assume that sensors in group  $G(i, j)$  are uniformly deployed within the zone  $Z(i, j)$ . However, we note that the sensor distribution pattern in the real-world may not follow any pattern due to manual distribution, helicopter drop, and so on. On the other hand, uniform sensor distribution is an ideal scenario and should be achieved as the goal via many deployment methods. Due to its simplicity and uniform density within the sensor deployment area, it significantly simplifies the analysis of many sensor network applications, such as environmental sensing, positioning, and so on. For the scenarios in which obstacles exist in the deployment area and irregular distribution patterns (cannot be simply described by one or several probability distribution functions) exist, simulation is the best way to study the proposed key management scheme. In this paper, we will assume uniform distribution and present an analytical approach for our proposed key management

scheme; consideration of irregular distribution (as well as which kinds to consider) and its study through simulation is not considered part of this paper and will be investigated separately and reported elsewhere.

#### 4.4 Phase 4: Key Discovery

The purpose of key discovery is to set up a *local key graph* for each sensor (see section 3.1, phase 4; and the local key graph example shown in Fig. 3). After deployment, each sensor, say  $[(i, j), u]$ , initiates this phase by broadcasting its identifier  $[(i, j), u]$  and the list of the key spaces selected for sensor  $[(i, j), u]$ . If two sensors have selected the same key space, they can establish the *direct key*.

After receiving all neighbors' *ids* and establishing the *direct keys*, sensor  $u$  builds a *local key graph* that contains all its neighbors as vertices. In a *local key graph*, a logical link is created between  $u$  and any of its neighbors if they share a *direct key*. Then, sensor  $u$  creates a neighbor list that contains all the neighbors that the *direct keys* have been set up between them. The neighbor list is then encrypted by the *direct keys* and sent to all the nodes in  $u$ 's neighbor list. When a neighbor  $v$  receives the neighbor list, it repeats the process, i.e.,  $v$  encrypts the neighbor list by using the *direct key* shared between  $v$  and each of nodes in  $v$ 's neighbor list, and then the encrypted neighbor list is sent out. This process is continued until a node discovers that it is not a neighbor of the original node  $u$ ; then, the node drops the neighbor list. After receiving the neighbor list, if  $v$  is both listed in  $u$ 's neighbor list and the receiver's neighbor list, the receiver updates its *local key graph* by adding a link between the source node  $u$  and the node  $v$ ; and after receiving all the neighbors' lists and updating the *local key graph*, the sensor constructs its *local key graph* that will be used in pairwise key establishment phase to set up *indirect keys*.

#### 4.5 Phase 5: Pairwise Key Establishment Protocol

Our proposed PKE protocol consists of two sub-phases: 1) the key establishment within a zone, 2) the key establishment between adjacent zones.

—*Key establishment within a zone:*

For key establishment within the same zone, using the derived *local key graph* in the key discovery phase, a sensor can use *source routing* [Huang et al. 2005], by explicitly specifying the key path (by listing hops), to send requests and then to establish *indirect keys*. The number of neighbors located within the same zone of a node is determined by the location of the node. Figure 9(a) shows the number of neighbors within the same zone for a sensor located in  $Z(i, j)$ . We need to have some guarantee that a sensor can establish pairwise keys with the majority of its neighbors; using the probability that a sensor can set up pairwise keys with  $n$  of its neighbors within  $h$  hops (for details, refer to [?]), we can determine how many keys are required to be preinstalled in a sensor.

—*Key establishment between adjacent zones:*

After the first phase of key establishment, the system goes into the second phase of the key establishment process to set up *indirect keys* with nodes located in the adjacent zones. We assume that sensors have established pairwise keys with their neighbors in the same zone. When a sensor wants to set up *indirect keys*

Table II.  $t$ -faulty resilient multi-path key establishment scheme.

preinstalled secrets for each sensor	
generator polynomial $g(x)$ , $2t$ roots $\alpha_1, \dots, \alpha_{2t}$ , where $n - k = 2t$ , see Algorithm B.1 in Appendix B	
key establishment procedure, $(n, k)$ RS codes, multi-path scheme	
sender $u$	receiver $v$
1. generates $p$ node-disjoint paths between $u$ and $v$	1. uses majority rule to eliminate bad codeword(s)
2. generates key message polynomial $m(x)$ , see Algorithm B.1	2. composes received key polynomial $r(x)$
3. creates $k$ codewords $m'_i$ , $i = 1, \dots, k$ , see Algorithm B.1	3. uses Algorithm B.2 step 2 to identify faulty path(s)
4. uses source routing to send at most $t$ codewords on each path where $t = (n - k)/2$	4. uses Forney's algorithm to derive error polynomial $e(x)$
	5. recovers the original message polynomial, see Algorithm B.2 step 3
properties of proposed multi-path key establishment scheme	
1. resilient to $t = (n - k)/2$ faulty paths when $(n, k)$ Reed-Solomon codes are used.	
2. receiver can identify the faulty key establishment paths.	
3. no interactive communications are required, which is communication efficient.	
4. Reed-Solomon error correcting codes are computationally efficient: in the order of $O(n \log^2 n)$ , see [Sarwate 1977].	

with its neighbors in the adjacent zones, it broadcasts the desired node-list. A neighbor of the requestor within the same zone who already shares a *direct key* with the nodes in the requestor's list acts as a proxy and does the following: 1) it selects an *indirect key* for the pair, 2) it encrypts the selected *indirect key* by using the pairwise keys already set up between itself and the requestor and the *direct key* already shared between itself and the destination node, 3) it sends the two encrypted messages to the requestor. Upon receiving the response, the requestor forwards the encrypted pairwise key to the destination node. Note that during the first phase of PKE, nodes have already set up pairwise keys to all their neighbors within the same zone; thus, during the second phase of PKE, as long as there exists one node that has a link to the neighboring zone, it then can be used as a bridge to set up pairwise keys to the neighboring zone for all its neighbors. For performance assessments on PKE protocol, refer to section 8.

**4.5.1 Multi-path pairwise key establishment.** During the pairwise key establishment phase, an intermediate node may have already been compromised, i.e., it has become an inside attacker. Basically, the target of insider attack is the PKE protocol. The attacker wants to prevent the sensors from establishing *indirect keys* or sniffing *indirect keys* established among uncompromised nodes. The insider node can behave just like a normal node and record all passing-through information; this type of an insider attack is called a *passive key-establishment attack*. Note that the insider node can also change, drop, or forge PKE messages to malfunction PKE protocol; this type of attack is called an *active key-establishment attack*.

In order to counter insider attacks, multipath pairwise key establishment scheme has been proposed [Chan et al. 2003; Zhu et al. 2003]. However, these schemes are vulnerable under *active key-establishment attacks* (see discussion in Section 2).

To counter active insider attacks, such as stop forwarding and cheating attacks, we propose a multi-path pairwise key establishment scheme by using multiple node-disjoint paths and Reed-Solomon error correcting codes [Reed and Solomon 1960]. The properties and operational procedures of the proposed scheme are shown in Table II. The related algorithms are given in Appendix B.

A sensor applies the Reed-Solomon encoding scheme to partition an *indirect key* seed into  $n$  codewords<sup>4</sup>. Each codeword is transmitted via a different node-disjoint path. Note that on each hop, the keyword is encrypted by a pairwise key. The receiver applies the Reed-Solomon decoding scheme to identify the faulty key establishment paths and then recovers the original *indirect key* seed. The receiver can derive the *indirect key* based on correctly receiving at least  $k$  codewords, where  $k \leq n$ . The main benefits of our proposed scheme are as follows:

- The proposed scheme is resilient to  $t$  faulty paths. If  $(n, k)$  Reed-Solomon codes are used, then  $t = (n - k)/2$ .
- The receiver can identify faulty key establishment paths.
- No interactive communications are required to identify the faulty key establishment paths, which is communication efficient.

The security and performance analysis of our proposed multi-path pairwise key establishment protocol is presented in Section 7.

## 5. SECURITY PERFORMANCE ANALYSIS I: AGAINST SELECTIVE NODE CAPTURE ATTACK

In selective node capture attacks, the attacker targets at compromising the communication links among uncompromised sensors. The consequence of this attack is the exposure of information transmitted on the compromised communication links. To achieve this goal, the attacker can mount a node capture attack on a deployed sensor and read secret information (keys) from its memory. These keys may be used among uncompromised sensors, thus we use the fraction of compromised links among uncompromised sensors to evaluate our scheme.

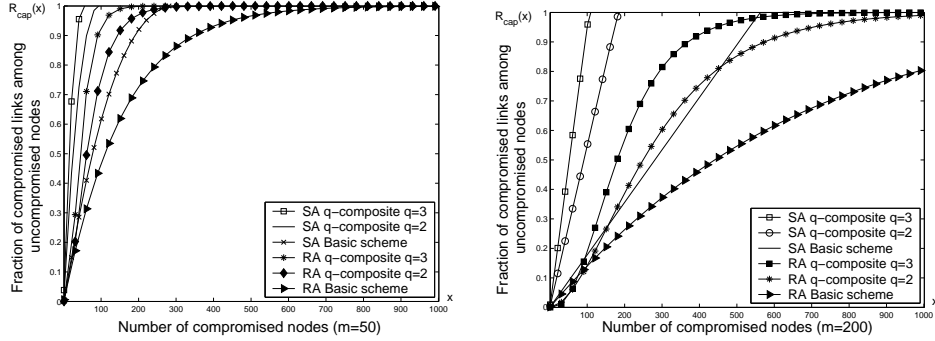
In our scheme (described in Section 4.2), we limit the number of distributed key rows in the key matrix to  $\lambda$ . Thus, no matter how many sensors are captured by attackers, they cannot derive the pairwise keys used between uncompromised sensors. We compare our scheme with existing RKP approaches. We use  $R_{cap}(x)$  to represent the fraction of compromised links among uncompromised nodes, where  $x$  is the number of compromised nodes. Thus,  $1 - R_{cap}(x)$  represents the resilience to selective node capture attacks.

We model the selective attack to the P-RKP scheme by using a heuristic technique (described below). Below,  $C_x$  is the cardinality of the set of compromised keys when  $x$  nodes are compromised, and  $i$  is a variable. We use  $B$  to represent the value that an attacker uses to inspect and then to decide which sensor to capture next. Then:

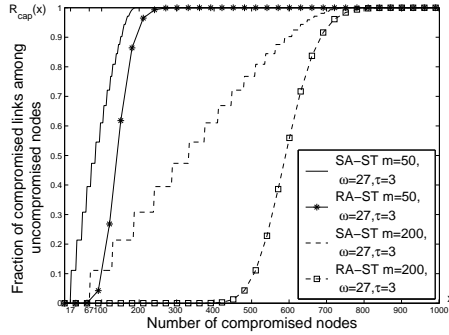
$$B = \frac{\binom{P-C_x}{m-i} \binom{C_x}{i}}{\binom{P}{m}} (N - x), \quad (i = 0, \dots, m), \quad (5)$$

where  $\frac{\binom{P-C_x}{m-i} \binom{C_x}{i}}{\binom{P}{m}}$  is the probability that there exists uncompromised nodes and each of them has  $m - i$  keys not already compromised;  $N - x$  is the total number of uncompromised nodes in the system.

<sup>4</sup>A sender uses  $(n, k)$  Reed-Solomon codes to partition a message to  $n$  codewords. The receiver can correctly recover the original message if he/she can correctly receive  $(n - k)/2$  number of codewords.



(a) Selective attack for P-RKP schemes ( $m=50$ ). (b) Selective attack for P-RKP schemes ( $m=200$ ).



(c) Selective attack for SK-RKP schemes.

Fig. 5. Selective attack for RKP schemes.

The heuristic method is as follows. Initially, when  $i = 0$ , an attacker can arbitrarily capture a sensor and derive  $m$  keys preinstalled in the captured sensor and then,  $C_x = m$ . Then, he inspects  $B$ : if  $B \geq 1$ , he continuously captures the nodes with  $m - i$  keys that are not already compromised, and for each capture,  $C_x$  is increased by  $m - i$ ; if  $B < 1$ , he increases  $i$  by 1 until  $B \geq 1$ . He then captures the sensors with  $m - i$  keys that are not already compromised. The attacker continues this process until the condition  $m = i$  is fulfilled or the entire key pool is compromised. The condition  $B > 1$  means there exists uncompromised sensors that have  $m - i$  keys which are not already compromised. Figure 5(a) and 5(b) show the comparison between the *selective-node-capture attack* (SA) and *random-node-capture attack* (RA) to P-RKP schemes when  $m = 50$  and  $m = 200$ . The comparative studies show that the selective node capture attack can gain more information than the random node capture attack with the same number of captured sensors.

In the SK-RKP scheme, the attacker can selectively capture sensors that possess keys within the same key space. Once  $\lambda + 1$  sensors in a key space are compromised, all the keys in that key space are compromised. In this fashion, an attacker can incrementally capture the sensors that use the same key space. Since sensors possess keys from more than one key space, the number of sensors required to be captured

to compromise subsequent key spaces is less. We use  $c(i)$  to represent the average number of additional sensors to be captured in order to compromise a key space when  $i - 1$  key spaces are already compromised. In order to compromise the first key space, the attacker needs to capture at least  $\lambda + 1$  nodes (i.e.  $c(1) = \lambda + 1$ ). Since each sensor is allocated  $\tau$  key spaces ( $\tau \geq 2$ ), a captured node also uses an uncompromised key space with probability  $p' = \frac{\tau-1}{\omega-1}$ . Thus, to compromise  $i^{th}$  key space, we have

$$c(i) = \lambda + 1 - \sum_{k=1}^{i-1} c(k) \cdot p', \quad 2 \leq i \leq \omega. \quad (6)$$

In Figure 5(c), we use (6) to show SK-RKP schemes [Liu and Ning 2003a; Liu et al. 2005; Du et al. 2003; Du et al. 2005] under the selective node capture attack (SA) and the random node capture attack (RA). As shown in the figure, the threshold of the SK-RKP scheme decreases dramatically under the selective node capture attack. In the figure, the threshold values under SA are: 17 with  $\omega = 27, \tau = 3$ , and  $m = 50$ ; 67 with  $\omega = 27, \tau = 3$ , and  $m = 200$ .

In summary, the security performance under the selective node capture attack of existing RKP and SK-RKP approaches drops dramatically comparing to the security performance under random node capture attack. However, using our area partitioning and multi-group deployment approaches (as described in Section 4), we can completely prevent any selective node capture attacks, i.e., the attackers cannot compromise any pairwise keys that are used between uncompromised sensors (the performance lines of our scheme should be  $R_{cap}(x) = 0$ ).

## 6. SECURITY PERFORMANCE ANALYSIS II: AGAINST NODE FABRICATION ATTACK

The compromised sensors cannot be easily detected since they can set up pairwise keys with uncompromised sensors<sup>5</sup>. The sensor network is vulnerable to this attack due to lack of authentication mechanism for large-scale sensor networks. This attack is more severe as compared to just replicating captured nodes as the attacker may have enough information to fabricate sensors with more than one identity with a single compromised sensor.

If there is a unique and one-to-one verifiable connection between node  $id$  and preinstalled keys, the preinstalled keys can be used as the authentication keys among sensors. In this section, we will conduct a comparative study of resilience to node fabrication attacks. The comparative candidates are the key discovery approaches for P-RKP schemes proposed in [Di Pietro et al. 2004] and [Mehta et al. 2005] and our SK-RKP solution.

### 6.1 Fabricate Compromised Nodes – Nodes Replication

The attacker compromises a sensor and clones the sensor. In fact, the attacker does not need to clone many sensors but deploy Sybil attack [Douceur ] by fabricating sensor  $ids$ . The Sybil attack is equivalent to deploying many cloned sensors; note

<sup>5</sup>Several anomaly detection approaches [Parno et al. 2005; Du et al. 2006; Liu et al. 2005a] have been recently proposed to malicious sensors. However, this paper does not focus on the anomaly detection.

that in this paper, we do not differentiate the difference between a Sybil attack and a node-fabrication attack. Due to lack of *a-priori* knowledge of post-deployment configuration, uncompromised sensors cannot detect the cloned sensor as an anomalous sensor. This attack can have more severe consequences as compared to the passive listening attacks on communication links between uncompromised nodes, since the attacker can implant malicious codes in replicated sensors to malfunction the PKE protocol. This is a typical form of an insider attack which will be discussed further in Section 7.

## 6.2 Fabricate Uncompromised Nodes

In this attack, the attacker compromises only few sensors and uses the captured keys to fabricate sensors with new identities. Then, the attacker can deploy the fabricated nodes in the network. The uncompromised sensors in the network cannot detect the fabricated nodes as anomalous nodes as long as they can set up pairwise keys with them. This attack is more severe as compared to passive eavesdropping attack as the attacker may have enough information to fabricate many sensors with many different identities and possibly outnumber the original set of sensors.

**6.2.1 Node fabrication attack to P-RKP scheme.** The attacker can launch the node fabrication attack on the P-RKP scheme by capturing only a few sensors. Since there is no identity authentication in the P-RKP scheme [Eschenauer and Gligor 2002; Chan et al. 2003], by capturing two nodes, the attacker can fabricate and deploy  $\binom{2m}{m}$  new nodes without being detected. These fabricated nodes are apparently good nodes, since they all have valid keys. Thus, the fabricated nodes can quickly outnumber the uncompromised nodes. To mitigate node fabrication attack, Di Pietro et al. [Di Pietro et al. 2004] and Mehta et al. [Mehta et al. 2005] proposed using a one-way function to distribute and discover the shared keys. The basic idea behind their approaches is to use a node *id* as the input to a one-way function multiple times and the generated chain values modulo the size of key pool can be used to identify the key *id* in the key pool. Using an one-way function to identify the keys can be used in both the key predistribution phase and key discovery phase. Mehta et al. [Mehta et al. 2005] use simulation to prove this method is equivalent to randomly selecting keys from a large key pool. In this way, a connection is built between the node *id* and preinstalled keys via the one-way function. As a result, attackers cannot arbitrarily fabricate sensors since they must possess the preinstalled keys that are identified by the hash-chain values.

Although one-way function based key discovery approaches can mitigate the node fabrication attack, the attackers can still fabricate sensors without being detected. To fabricate a sensor, the attacker must compromise enough sensors to find a node *id* that can be used to set up communication links with uncompromised nodes. In order to connect to the network via an uncompromised node, the fabricated node needs to satisfy the following conditions: (a) it should share the required number of keys with the uncompromised node when *q*-composite scheme is used; (b) if the condition (a) is satisfied, all shared keys not only must be known to the attacker, but also they must be in the right sequence. The probability that a fabricated node

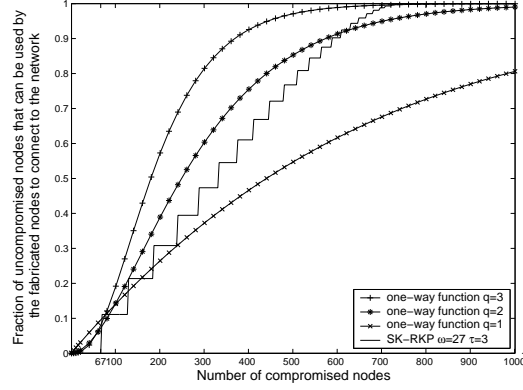


Fig. 6. Node fabrication attack to one-way function key discovery approach and SK-RKP approach:  $m = 200$ ,  $\omega = 27$ ,  $\tau = 3$ ,  $p_1 = 0.3$ .

satisfies the above two conditions with  $x$  captured nodes is computed as:

$$p_f(x) = \frac{1}{p_{connect}} \sum_{i=q}^m \frac{\binom{P}{m} \binom{m}{i} \binom{P-m}{m-i}}{\binom{P}{m}^2} \left(\frac{C_x}{P}\right)^i, \quad (7)$$

where  $\sum_{i=q}^m \frac{\binom{P}{m} \binom{m}{i} \binom{P-m}{m-i}}{\binom{P}{m}^2}$  is the probability that  $i$  keys are shared between two nodes,  $C_x = [1 - (1 - \frac{m}{P})^x]P$  is the number of keys compromised due to capture of  $x$  nodes,  $x$  is the number of captured nodes, thus  $(C_x/P)^i = ([1 - (1 - \frac{m}{P})^x])^i$  is the probability that all  $i$  keys are compromised due to  $x$  captured nodes, and  $p_{connect}$  is the probability of node connectivity for  $q$ -composite schemes given in (8):

$$p_{connect} = p(q) + p(q+1) + \dots + p(m), \quad (8)$$

where  $p(q)$  is the probability that two nodes have exactly  $q$  keys in common. The probability  $p(q)$  is presented in [Chan et al. 2003] and is given as:

$$p(q) = \frac{\binom{P}{q} \binom{P-q}{2(m-q)} \binom{2(m-q)}{m-q}}{\binom{P}{m}^2}.$$

Using (7) we draw the Figure 6. It shows the fraction of uncompromised nodes that can be used by fabricated nodes to connect to the system when  $q = 1, 2, 3$ . Since the one-way function key discovery approach cannot totally prevent the attacker from fabricating new nodes and the derived shared key is not unique, we cannot use the *direct key* of the P-RKP scheme for authentication purpose.

**6.2.2 Node Fabrication Attack to the SK-RKP scheme.** The SK-RKP scheme is also vulnerable to the node fabrication attack. However, there are some restrictions for the attacker. First, the attacker is required to capture more than  $\lambda$  sensors in order to compromise a key space. Second, the attacker cannot arbitrarily generate new *ids* for the fabricated sensors, since the *ids* indicate the rows of the key matrix  $A$ . The wrong *id* cannot set up the pairwise key between the fabricated sensor and

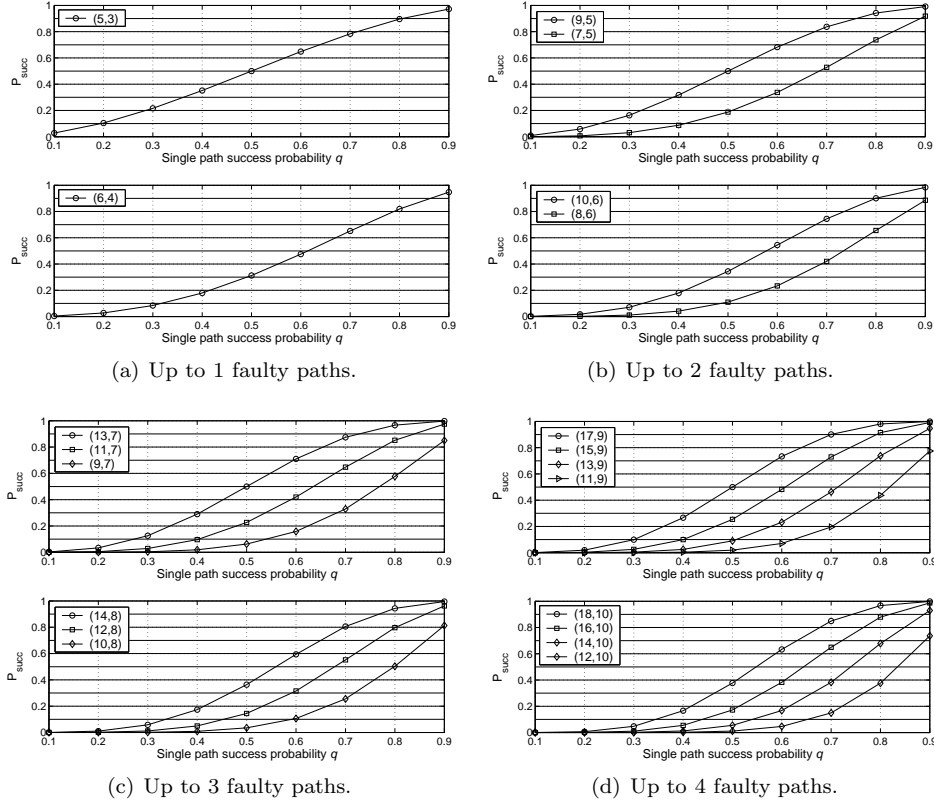


Fig. 7. Evaluate multi-path success probability  $P_{succ}$  given the number of faulty paths is up to 1, 2, 3, and 4.

the uncompromised sensor. If we restrict the number of distributed rows of a key matrix  $A$  to no more than  $\lambda$ , we can prevent the node fabrication attack. Even if the attacker can compromise all deployed sensors, he cannot derive new rows of any key matrix. The previous proposals [Liu and Ning 2003a; Liu et al. 2005; Du et al. 2003; Du et al. 2005] cannot fulfil this requirement due to large size of sensor networks. For example, if we stay with the restriction that no more than  $\lambda$  rows of any key space are allowed to be distributed to sensors, for given  $\omega$ ,  $\tau$ , and  $\lambda$ , the maximum number of sensors that can be deployed is  $\frac{\lambda \cdot \omega}{\tau}$ . We can easily derive that the supported network size  $n_z$  is linearly increased with respect to the increase of the  $\lambda$ .

### 6.3 Countermeasures to Node Fabrication Attack

In Figure 6, using the SK-RKP scheme, zero fraction of the network is compromised when  $x = 0 \sim 67$ . This property of the SK-RKP scheme allows us to design an authentication protocol for the key discovery phase. Recall from Section 6.2.2 that to restrict the number of sensors that are allowed to install keys from a given key space, the size of supported network is also restricted. Thus, if we partition a

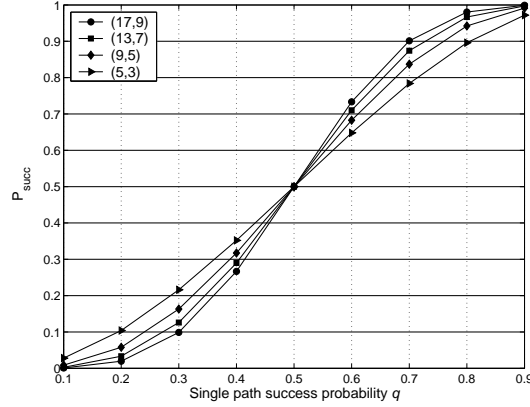


Fig. 8. Success probabilities for (17, 9), (13, 7), (9, 5), and (5, 3) RS multi-path coding schemes.

large sensor network into multiple small areas, we can utilize the perfect security property of the SK-RKP scheme within each zone.

Note that in Section 4, we have proposed to use a unique structured key pool for each zone and to restrict at most  $\lambda$  rows of a key space that are distributed to sensors. Our scheme is resilient to node fabrication attacks (including node replication attacks) due to the following two facts: (a) Since no more than  $\lambda$  key rows are distributed to sensors within a zone, attackers cannot fabricate uncompromised sensors. Thus, attackers can only replicate captured sensors. (b) Since the key matrix used for one zone is different from the other zones, the attacker cannot deploy the replicated sensors to the other zones. Thus, the replication attack is restricted within a relatively small area, i.e., the replicated nodes can only setup pairwise keys with uncompromised sensors within the zone where it was captured.

## 7. SECURITY PERFORMANCE ANALYSIS III: AGAINST INSIDER ATTACK

An important question that we want to answer is: “in what scenarios, do we want to use multi-path routing instead of single path routing?” In other words, “what is the condition when multi-path routing is preferred over single-path routing?” To see this, we need to evaluate the multi-path success probability  $P_{succ}$ , given the single path success probability  $q$  and the number of paths. We have found an evaluation model developed by Tsirigos and Hass [Tsirigos and Haas 2004] to be applicable to our scenario. When uniform codeword allocation is applied, we have:

$$P_{succ} = \sum_{i=k}^n \binom{n}{i} p^i (1-p)^{n-i}, \quad (9)$$

where  $(n, k)$  is the Reed-Solomon coding scheme and  $p$  is the probability that a receiver successfully receives a codeword from a path.

IN our analysis, we assume that each path has the same success probability  $q = 0.1, \dots, 0.9$ , and that exactly one codeword is sent through a path; we consider the number of faulty paths to be 1, 2, 3, and 4. In Figure 7(a)~Figure 7(d), we plot success probabilities  $P_{succ}$  of the total number of paths from 3 paths to 10 paths.

We have the following observations:

- Using Reed-Solomon codes, we can use the majority rule to filter out bad parity codes. Thus, A 4-path multi-path routing does not give any benefit compared to a 3-path multi-path routing. We observe similar behavior with 6 and 5 paths, 8 and 7 paths, 10 and 9 paths. This is since faulty nodes on different paths can collude to generate the same parity codes, and it can create uncertainty at the destination node.
- The success probability of the even number of paths is always lower than that for the corresponding odd number of paths. Thus, we use odd number of paths in our schemes.
- Using our  $(n, k)$  Reed-Solomon multi-path coding scheme, we observe that bigger the value of  $n$  results in  $P_{succ}$  to be higher when the number of paths  $k$  is fixed.

In order to decide whether to apply multi-path routing or single path routing, we plot the success probabilities for  $(17, 9)$ ,  $(13, 7)$ ,  $(9, 5)$ , and  $(5, 3)$  Reed-Solomon multi-path coding schemes in Figure 8. When  $P_{succ} \geq q$ , the performance of multi-path routing is better than single path routing. For example, if the  $(5, 3)$  Reed-Solomon coding scheme is used, we observe that the success probabilities  $q$  of single-path routing are always greater than the corresponding multi-path success probability  $P_{succ}$  when  $q < 0.5$ . The same behavior is observed in the cases of  $(17, 9)$ ,  $(13, 7)$ ,  $(9, 5)$ , and  $(5, 3)$  Reed-Solomon multi-path coding schemes. Thus, in the case of uniform block allocation and uniform success probability distribution, we conclude that the multi-path routing is preferred to single path routing when the number of paths increases and path success probability  $q > 0.5$ .

It may be noted that the adversaries can intentionally capture sensors located at area borders to compromise the PKE procedures. However, based on our analysis in Section 8.1.2, the border sensors can also have high probabilities to set up multiple 2-hop paths to the neighbor zone. In addition, our proposed multi-path PKE scheme can relieve the insider attacks due to compromised nodes.

## 8. OPERATIONAL PERFORMANCE ASSESSMENTS

### 8.1 Local Key Graph Connectivity

We now present the *Local Key Graph* connectivity analysis for sensors located within the same zone and in adjacent zones. Our analysis is based on the following assumptions:

- The area covered is a two-dimensional Cartesian plane. A zone is represented by an area  $x \in [0, a], y \in [0, a]$ , where  $(x, y)$  is a point in the two-dimensional Cartesian plane.
- All sensors have equal communication radius,  $R$ , and hence cover the same size of area, where  $R \leq a/2$ .
- The sensors are uniformly distributed in a deployment region and the average number of neighbors for each sensor is  $n'$ . The density of the deployed sensors is  $\rho = \frac{n'}{\pi R^2}$ .

According to the assumption presented above, the number of deployed sensors within each zone is  $n_z = a^2 \rho \approx \left\lceil \frac{a^2 n'}{\pi R^2} \right\rceil$ . For our analysis, we consider  $R = 40$

meters, and  $a = 100$  meters.

#### Sensor Coverage – within the Same Zone

We first consider the coverage of sensor  $[(i, j), b]$  in its zone  $Z(i, j)$ . Given a position  $(x, y)$  for sensor  $[(i, j), b]$ , the sensor coverage is given as follows:

$$\mathcal{C}_b(i, j)|_{(x, y)} = \begin{cases} \mathcal{C}_b^1(i, j)|_{(x, y)} & , 0 \leq \sqrt{x^2 + y^2} \leq R; \\ \mathcal{C}_b^2(i, j)|_{(x, y)} & , R < \sqrt{x^2 + y^2} \leq a/2. \end{cases}$$

The expressions for  $\mathcal{C}_b^1(i, j)$  and  $\mathcal{C}_b^2(i, j)$ , along with the proofs are given in Appendix C. From these results, the number of neighbors of sensor  $[(i, j), b]$  within the zone  $Z(i, j)$  can be obtained as:

$$\mathcal{N}_b(i, j) = \rho \cdot \mathcal{C}_b(i, j), \quad (10)$$

where  $\mathcal{N}_b(i, j)$  is the number of neighbors of sensor  $[(i, j), b]$  within the zone  $Z(i, j)$ . In Figure 9(a), we show the contour curves of the average number of neighbors of sensor  $[(i, j), b]$  within the zone  $Z(i, j)$ .

#### Sensor Coverage – in Different Zone

In Figure 9(a), we show that there are 8 possible zones surrounding zone  $Z(i, j)$ . We use superscripts  $+$  and  $-$  to represent the area coverage and sensor coverage between two neighboring zones. For example  $\mathcal{C}_b(i^+, j^-)$  and  $\mathcal{N}_b(i^+, j^-)$  represent the area coverage and sensor coverage of sensor  $[(i, j), b]$  in zone  $Z(i + 1, j - 1)$ . Similarly,  $\mathcal{C}_b(i, j^-)$  and  $\mathcal{N}_b(i, j^-)$  represent the area coverage and sensor coverage of sensor  $[(i, j), b]$  in zone  $Z(i, j - 1)$ . The number of neighbors that node  $[(i, j), b]$  covers in a neighboring zone is given as:

$$\mathcal{N}_b(i^*, j^*) = \rho \cdot \mathcal{C}_b(i^*, j^*), \quad (11)$$

where  $*$  represents  $-$ ,  $+$ , or none. The representations and proofs of neighboring zone coverage  $\mathcal{C}_b(i^*, j^*)|_{(x, y)}$  are given in Appendix D.

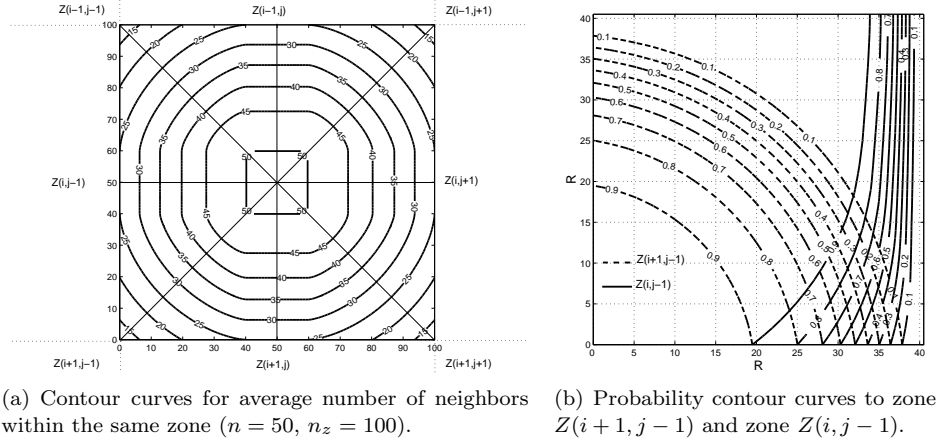
**8.1.1 Local Key Graph Connectivity within the Same Zone .** The number of keys preinstalled in each sensor is represented by  $m$ . According to the deployment pattern shown in Figure 4, we select a unique key pool for each zone, i.e.,  $\mathcal{P}(i, j)$  for  $Z(i, j)$ . To determine the size of key pool (i.e.,  $|\mathcal{P}(i, j)|$ ) and the number of keys selected (i.e.,  $m = (\lambda + 1)\tau$  and will be discussed in section 8.3) from the key pool for sensor  $[(i, j), b]$ , we use the equations of P-RKP scheme proposed by Eschenauer and Gligor [Eschenauer and Gligor 2002] and further modified for SK-RKP scheme by Du et al. [Du et al. 2003]:

$$p_1 = 1 - \frac{\binom{\omega}{\tau} \binom{\omega - \tau}{\tau}}{\binom{\omega}{\tau}^2} = 1 - \frac{((\omega - \tau)!)^2}{(\omega - 2\tau)! \omega!}, \quad (12)$$

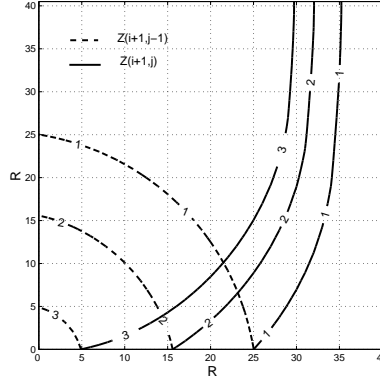
where  $p_1$  is the probability that given two sensors share at least one key.

For sensor  $[(i, j), b]$  in zone  $Z(i, j)$ , the number of neighbors within its zone is  $\mathcal{C}_b(i, j)$ . As shown in Figure 9(a), if the average number of neighbors of a sensor,  $n'$ , is 50, the zone has total of  $n_z = (n'a^2)/(\pi R^2)$  sensors and there are approximately 11 nodes in the zone with less than 25 neighbors from the same zone. If we assume the number of neighbors of a sensor is 25, using the *local key graph*<sup>6</sup> connectivity

<sup>6</sup>Here, the local key graph is composed by the sensors within the same zone.



(a) Contour curves for average number of neighbors within the same zone ( $n = 50$ ,  $n_z = 100$ ). (b) Probability contour curves to zone  $Z(i+1, j-1)$  and zone  $Z(i, j-1)$ .



(c) Connectivity contour curves to the neighboring zone  $Z(i+1, j-1)$  and zone  $Z(i, j-1)$  with  $q=1, 2, 3$ . The probability  $p_{\bar{q}}(i^*, j^*)$  to share at least  $q$  keys is 0.8, see (14).

Fig. 9. In this example, sensors are deployed in a  $100 \times 100$  square meters area and all sensors have the same transmission radius (40m).

presented in [?], we derive the probability  $p_1 = 0.5$ . When  $p_1 \geq 0.5$ , the *local key graph* is connected with probability greater than 0.996 within three hops. In the worst case, the sensor is located at the corner of the square area, and has approximately 12 neighbors within the same zone. In this case, the probability that the *local key graph* is connected within five hops is 0.8736.

**8.1.2 Local Key Graph Connectivity between Two Adjacent Zones.** The node  $[(i, j), b]$  may be located close to the boundary of two neighboring zones,  $Z(i, j)$  and  $Z(i^*, j^*)$ . The number of neighbors of node  $[(i, j), b]$  located within these two zones can be represented by  $\mathcal{N}_b(i, j)$  and  $\mathcal{N}_b(i^*, j^*)$ . Node  $[(i, j), b]$  is considered to be connected to the neighboring zone as long as it can find at least one neighbor,

$b'$ , located in  $\mathcal{C}_b(i, j)$  who shares a key with at least one of nodes,  $b''$ , located in  $\mathcal{C}_b(i^*, j^*)$ . Using (10) and (11), we can derive the probability  $p(i^*, j^*)$  that sensor  $[(i, j), b]$  can connect to the neighboring zone with the help of all its neighbors.

$$p(i^*, j^*) = 1 - \frac{\binom{n_z - \mathcal{N}_b(i, j)}{\mathcal{N}_b(i^*, j^*)} \binom{n_z - \mathcal{N}_b(i^*, j^*)}{\mathcal{N}_b(i, j)}}{\binom{n_z}{\mathcal{N}_b(i^*, j^*)} \binom{n_z}{\mathcal{N}_b(i, j)}}. \quad (13)$$

Note that (10) and (11) are derived from  $\mathcal{C}_b(i, j)$  and  $\mathcal{C}_b(i^*, j^*)$ , which are the functions of two-dimensional Cartesian coordinates with the position  $(x, y)$ . Thus  $p(i^*, j^*)$  is the function of  $(x, y)$ . Using the (13), we draw the probability contour curves in Figure 9(b), where a node in  $Z(i, j)$  can connect to its neighboring zones  $Z(i, j^-)$  and  $Z(i^+, j^-)$  with parameters  $a = 100m, R = 40m, n' = 50, n_z = 100$ .

**8.1.3 Multi-path Connectivity between Neighboring Zones.** Figure 9(b) shows the contour curves of the probabilities that a node in  $Z(i, j)$  can connect to its neighboring zones  $Z(i, j - 1)$  and  $Z(i + 1, j - 1)$ . The probability that a node can connect to neighboring zones with the help of exactly  $k$  neighbors is given as follows:

$$p_k(i^*, j^*) = \frac{\binom{n_z}{k} \binom{n_z - \mathcal{N}_b(i, j) - k}{\mathcal{N}_b(i^*, j^*) - k} \binom{n_z - \mathcal{N}_b(i^*, j^*) - k}{\mathcal{N}_b(i, j) - k}}{\binom{n_z}{\mathcal{N}_b(i^*, j^*)} \binom{n_z}{\mathcal{N}_b(i, j)}}.$$

The probability that a node can connect to neighboring zones with the help of at least  $q$  neighbors is denoted by  $p_q(i^*, j^*)$  and is given as follows:

$$p_q(i^*, j^*) = 1 - [p_0(i^*, j^*) + \dots + p_{q-1}(i^*, j^*)]. \quad (14)$$

Figure 9(c) shows the range in which a sensor can connect to its neighboring zones with at least  $q$  links via its neighbors, where  $q = 1, 2, 3$  and the connectivity probability is 0.8. Thus, a sensor can randomly select  $q$  neighbors that respond to the requests and send the responses to the destination nodes. The selected  $q$  destination nodes can help the sensor set up  $q$  paths to any of the neighbors in the adjacent zone. Thus, we can apply our proposed multi-path PKE scheme (see section 4.5.1) to set up the pairwise keys.

Comparing Figure 9(a) and Figure 9(c), almost all sensors that have less than 20 neighbors and some of sensors that have less than 25 neighbors can set up at three connections to the diagonal neighboring zones. The sensors that have less than 35 neighbors within the same zone may set up at three connections to the horizontal and vertical neighboring zones.

## 8.2 Communication Overhead Analysis

Here, we will derive the mathematical expressions for the probability that a sensor can set up key paths with all its neighbors within  $h$  hops in [?]. Since our key establishment procedure includes two phases, the key establishment within a zone and the key establishment between two adjacent zones, we analyze the communication overhead for each phase separately.

During the first phase of key establishment, the closer the sensor to the center of a zone, the smaller the communication overhead due to key establishment. For

example, for  $\tau = 2$ ,  $\omega = 7$ , the following table shows the number of hops and the corresponding *local key graph* connectivity probabilities: As shown in Figure 9(a),

Table III. The relation between the neighborhood size and the length of a key path.

# of neighbors	# of hops	<i>local key graph</i> connectivity
50	2	0.9957
40	2	0.9806
30	3	0.9996
25	3	0.9980
20	3	0.9893
15	4	0.9583

most pairwise keys can be set up within 3 hops. Since the communication overhead is proportional to the number hops of a path, the communication overhead is amplified due to multipath PKE scheme. As we can see in Table III, the denser the network, the shorter the key paths when the probability (i.e.,  $p_1$ ; see (12)) of sharing preinstalled keys is given.

During the second phase of key establishment, between two adjacent zones, as shown in Figure 9(c), a sensor can set up  $q$  paths to the neighboring zones with the probability of 0.8. Each path is a 2-hop path. A sensor applies  $(n, k)$  Reed-Solomon coding and partitions a pairwise key in  $q$  portions (where  $n = q$ ), and then sends them via  $q$  key paths. If the destination correctly receives  $l$  codewords where  $l \geq k$  or  $l \leq (n - k)/2$ , the pairwise key can be successfully set up, since all multipath to the neighboring zone is restricted by 2 hops. Thus, the communication overhead is only doubled compared to one-hop communications. In addition, the total communication overhead is also proportional to the number of paths.

### 8.3 Storage Overhead

Our proposed RKP-PKE scheme utilizes the key predistribution scheme proposed by Blom [Blom 1985] (for detailed description, see Appendix A). The scheme is built on two matrices: a publicly known matrix  $G$  of size  $(\lambda + 1) \times N$ ; a secret matrix  $D$  of size  $(\lambda + 1) \times (\lambda + 1)$  created by key distribution center. The matrix  $A$  of size  $N \times (\lambda + 1)$  is then created as  $A = (D \cdot G)^T$ . Each row of  $A$  is the keys distributed to a sensor and the row number can serve as a sensor's *id*. Our approach requires each sensor to select  $\tau$  key spaces (see section 4.2.1). Thus, a sensor is required to store  $m = (\lambda + 1)\tau$  keys that are used to set up pairwise keys within its zone, where  $\lambda$  is restricted by  $n_z = \lambda\omega/\tau$ . For example, if  $\tau = 2$ ,  $\omega = 7$ , and  $n_z = 100$ , then  $\lambda = 29$ . In addition to the keys selected from the key matrix  $A$ , each sensor is required to install at least one shared key with a unique sensor in each of its neighboring zones. The maximum number of neighboring zones is 8. Thus, the total number of keys that are needed to be preinstalled in a sensor is given as:

$$m = \left( \left\lceil \frac{n_z \tau}{\omega} \right\rceil + 1 \right) \tau + \gamma \alpha,$$

where  $\gamma$  is the number of neighboring zones and  $\alpha$  is the number of keys preinstalled for each pair of neighboring zones for a sensor. For all our analysis, we use the

following parameter setting:  $\gamma = 8$ ,  $\alpha = 1$ . Thus, the storage requirement for a sensor is  $m = 68$ .

Unlike the P-RKP scheme proposed in [Eschenauer and Gligor 2002] which requires  $m = 272$  to fulfil  $p_1 = 0.5238$ , our scheme requires  $m = 68$  which is much less. For the scheme specified in [Du et al. 2004], to achieve the  $p_1 = 0.5238$ , it requires 72 keys preinstalled for each sensor, which is a marginally higher than our scheme.

Our approach is similar to the group key based solution proposed by Liu et al. [2005b]. In [Liu et al. 2005b], the key predistribution is based on a two-dimensional group structure where two instances of key distributions  $D$  and  $D'$  are applied to the vertical structure and horizontal structure, respectively. Thus, the storage requirement equals to the summation of key storage required by both  $D$  and  $D'$ ; and we have  $|D| + |D'| = m$ , where  $m$  is the maximum storage allocated for storing keys. In our approach, we assume  $D$  is the SK-RKP scheme and  $D'$  is a deterministic key predistribution approach which requires the constant number of keys (i.e.,  $\leq 8$ ) to be installed; and we have the relation  $|D| + 8 = m$ . Since  $|D| \approx |D'|$  in [Liu et al. 2005b], our approach saves approximately 50% space when the group size is large. This advantage is due to the pre-known deployment information of our approach.

#### 8.4 Computational Overhead Analysis

The computational overhead arises mainly from the secure group key scheme introduced by SK-RKP scheme. In our schemes, we reduced the computational overhead significantly as compared to the SK-RKP schemes proposed in [Liu and Ning 2003a; Liu et al. 2005; Du et al. 2003; Du et al. 2005] that do not use the location information. For example, by using the SK-RKP scheme without using location information, for  $m = 200$ ,  $\tau = 2$ , and  $\lambda = 100$ , to derive a pairwise key, the total number of required modular multiplication operations is 200 (for the detail description of pairwise key establishment scheme refer to [Liu and Ning 2003a; Liu et al. 2005; Du et al. 2003; Du et al. 2005]). In our scheme, we only need to guarantee the local connectivity within a zone. We reduce the number of keys preinstalled in a sensor (see the analysis presented in section 8.3). If we restrict the number of sensors within a zone to  $n_z = 100$ , then  $\lambda = \lceil n_z \tau / \omega \rceil = 29$  where  $\omega = 7$ ,  $\tau = 2$ . Thus, the number of required modular multiplication operations to derive a pairwise key is only 58.

Finally, we note that public key cryptography algorithms (such as ECC and RSA) have been implemented in sensors. A general technique to improve the performance of public key algorithm is to reduce the Hamming weight of a multiplier (an exponent in RSA). According to [Koc 1994], the average number of multiplications involved by RSA scheme is  $\frac{3}{2}(k-1)$ , where  $k$  is the number bits of an exponent. Thus, for 1024-bit RSA scheme, it requires about 1535 modular multiplications. According to Gura et al. [2004], in a projective coordinate system, a point addition requires 9 multiplication and 5 squaring operations, and point doubling requires 4 multiplication and 4 squaring operations as the most expensive operations. If we assume that the average number of multiplications of a point addition requires 5 multiplications, for a 160-bit ECC point multiplication operation,  $\frac{3}{2}(k-1) \cdot 5 = 1193$  modular multiplications are required. Thus, public key based schemes involve more

computational overhead compared to our proposed approaches.

## 9. CONCLUSION

We have proposed a five phase RKP-PKE solution to counter various sensor network attacks. Our approach has benefitted from three schemes: the  $\lambda$ -restricted SK-RKP scheme, multiple group/zone sensor deployment, and a Reed-Solomon code-based  $k$ -path PKE protocol. We utilize the initial threshold of the SK-RKP scheme and the unique relation between key ring *id* and predistributed keys to guard against selective node capture and node fabrication attacks. RKP-PKE schemes are vulnerable to insider attack. We utilize the multiple groups/zones to restrict the insider attacks within a relative small range; and we use the multiple node-disjoint-path PKE protocol and fault-tolerance coding scheme (Reed-Solomon codes) to improve the system resilience to insider attacks.

RKP-PKE schemes have been studied extensively in recent years, but it is still in its infancy stage since many security and implementation problems have not been solved yet. For example, an insider attack is still the biggest threat for RKP-PKE schemes and no efficient solutions have been found to prevent it; a comprehensive study of multipath coding schemes is required; anonymity issues of wireless sensor networks; secure rekeying and node replenishment need to be studied; various sensor deployment methods—hybrid sensor distribution patterns—need to be studied; no PKE protocol prototype has been built and the security services of PKE protocol such as confidentiality, integrity, and authenticity are still unclear. It is hoped that this work can serve as an elicitation for researchers interested in further investigation into this vital area.

## REFERENCES

- AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., AND CAYIRCI, E. 2002. A Survey on Sensor Networks. *IEEE Communications Magazine* 40, 102–114.
- BLOM, R. 1985. An Optimal Class of Symmetric Key Generation Systems. In *EUROCRYPT'84*. Lecture Notes in Computer Science, vol. 209. Springer-Verlag, Paris, France, 335–338.
- BLUNDO, C., SANTIS, A. D., HERZBERG, A., KUTTEN, S., VACCARO, U., AND YUNG, M. 1998. Perfectly-Secure Key Distribution for Dynamic Conferences. *Information and Computation* 146, 1, 1–23.
- CHAN, H., PERRIG, A., AND SONG, D. 2003. Random Key Predistribution Schemes for Sensor Networks. In *Proceedings of 2003 Symposium on Security and Privacy*. IEEE Computer Society, Los Alamitos, CA, 197–215.
- DI PIETRO, R., MANCINI, L. V., AND MEI, A. 2004. Efficient and Resilient Key Discovery Based on Pseudo-Random Key Pre-deployment. In *Proceedings of 18th International Parallel and Distributed Processing Symposium (IPDPS)*. 217.
- DOUCEUR, J. R. The Sybil Attack. In *Proceedings of First International Workshop on Peer-to-Peer Systems (IPTPS)*. 251–260.
- DU, W., DENG, J., HAN, Y. S., CHEN, S., AND VARSHNEY, P. K. 2004. A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge. In *Proceedings of IEEE Information Communications Conference (INFOCOM)*. 586–597.
- DU, W., DENG, J., HAN, Y. S., VARSHNEY, P., KATZ, J., AND KHALILI, A. 2005. A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. *ACM Transactions on Information and System Security* 8, 2, 228–258.
- DU, W., DENG, J., HAN, Y. S., AND VARSHNEY, P. K. 2003. A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. In *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*. 42–51.

- DU, W., FANG, L., AND NING, P. 2006. LAD: Localization Anomaly Detection for Wireless Sensor Networks. *Journal of Parallel and Distributed Computing* 66, 7, 874–886.
- ESCHENAUER, L. AND GLIGOR, V. D. 2002. A Key-management Scheme for Distributed Sensor Networks. In *Proceedings of 9th ACM Conference on Computer and Communication Security (CCS-02)*. 41–47.
- GANESAN, D., GOVINDAN, R., SHENKER, S., AND ESTRIN, D. 2001. Highly-resilient, Energy-efficient Multipath Routing in Wireless Sensor Networks. *Mobile Computing and Communications Review* 4, 5, 11–25.
- GURA, N., PATEL, A., AND WANDER, A. 2004. Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. In *Proceedings of the 2004 Workshop on Cryptographic Hardware and Embedded Systems (CHES)*. 119–132.
- HUANG, D. AND MEDHI, D. 2005. A Byzantine Resilient Multi-path Key Establishment Scheme and Its Robustness Analysis for Sensor Networks. In *5th IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks*. 240b.
- HUANG, D., MEHTA, M., AND MEDHI, D. 2005. Source Routing Based Pairwise Key Establishment Protocol for Sensor Networks. In *Proceedings of 24th IEEE International Performance Computing and Communications Conference*. 177–183.
- HUANG, D., MEHTA, M., MEDHI, D., AND LEIN, H. 2004. Location-aware Key Management Scheme for Wireless Sensor Networks. In *Proceedings of ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)*. 29–42.
- KAHN, J. M., KATZ, R. H., AND PISTER, K. S. J. 1999. Next Century Challenges: Mobile Networking for "Smart Dust". In *Proceedings of International Conference on Mobile Computing and Networking (MOBICOM)*. 271–278.
- KARLOF, C. AND WAGNER, D. 2003. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. *Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols* 1, 2–3 (September), 293–315.
- KOÇ, C. K. 1994. High-speed RSA implementation. Tech. rep., RSA Laboratories.
- LINDSEY, S. AND RAGHAVENDRA, C. S. 2002. Pegasis: Power Efficient Gathering in Sensor Information Systems. In *Proceedings of IEEE Aerospace Conference*. 1125–1130.
- LIU, D. AND NING, P. 2003a. Establishing Pairwise Keys in Distributed Sensor Networks. In *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*. 52–61.
- LIU, D. AND NING, P. 2003b. Location-based Pairwise Key Establishments for Static Sensor Networks. In *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks (CCS'03)*. 72 – 82.
- LIU, D. AND NING, P. 2005. Improving Key Pre-Distribution with Deployment Knowledge in Static Sensor Networks. *to appear in ACM Transactions on Sensor Networks* 2, 204–239.
- LIU, D., NING, P., AND DU, W. 2005a. Detecting Malicious Beacon Nodes for Secure Location Discovery in Wireless Sensor Networks. In *Proceedings of the The 25th International Conference on Distributed Computing Systems*. 609–619.
- LIU, D., NING, P., AND DU, W. 2005b. Group Based Key PreDistribution in Wireless Sensor Networks. In *Proceedings of the ACM Workshop on Wireless Security (WiSe)*. 11–20.
- LIU, D., NING, P., AND LI, R. 2005. Establishing Pairwise Keys in Distributed Sensor Networks. *ACM Transactions on Information and System Security* 8, 1, 41 – 77.
- MANJESHWAR, A. AND AGRAWAL, D. P. 2001. TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. In *Proceedings of 15th International Parallel and Distributed Processing Symposium (IPDPS) Workshops*. 30189a.
- MEHTA, M., HUANG, D., AND HARN, L. 2005. RINK-RKP: A Scheme for Key Predistribution and Shared-key Discovery in Sensor Networks. In *Proceedings of 24th IEEE International Performance Computing and Communications Conference*. 193–197.
- PARNO, B., PERRIG, A., , AND GLIGOR, V. 2005. Distributed Detection of Node Replication Attacks in Sensor Networks. In *Proceedings of the IEEE Symposium on Security and Privacy*. 49–63.

- POTTIE, G. J. AND KAISER, W. J. 2000. Wireless Integrated Network Sensors. *Communications of the ACM* 43, 5 (May), 51–58.
- REED, I. S. AND SOLOMON, G. 1960. Polynomial Codes Over Certain Finite Fields. *SIAM Journal of Applied Math* 8, 300–304.
- SARWATE, D. V. 1977. On the Complexity of Decoding Goppa Codes. *IEEE Transactions on Information Theory* 23, 4, 515–516.
- SPENCER, J. H. 2001. *The Strange Logic of Random Graphs (Algorithms and Combinatorics)*. Springer Verlag.
- TSIRIGOS, A. AND HAAS, Z. J. 2004. Analysis of Multipath Routing Part I: The Effect on the Packet Delivery Ratio. *IEEE Transactions on Wireless Communications* 3, 1, 138–146.
- WICKER, S. B. 1995. *Error Control Coding for Digital Communication and Storage*. Prentice-Hall, NJ.
- ZHU, S., XU, S., SETIA, S., AND JAJODIA, S. 2003. Establishing Pair-wise Keys For Secure Communication in Ad Hoc Networks: A Probabilistic Approach. In *Proceedings of 11th IEEE International Conference on Network Protocols (ICNP)*. 326–335.

## Appendices

### A. STRUCTURED KEY-POOL RANDOM KEY PREDISTRIBUTION SCHEME

SK-RKP scheme is a modified version of P-RKP scheme. The main differences are the key pool structure and key discovery method. Unlike in P-RKP schemes, in SK-RKP scheme, each sensor is preloaded with a unique set of keys in its memory. The key discovery is not simply finding a shared key with the neighboring sensor, but using a set of polynomial variables (constructed by the keys possessed by the sensor) to derive the shared key. In addition, the key *id* can serve as the sensor *id* which is linked to the set of preinstalled keys.

The SK-RKP scheme uses the key predistribution scheme proposed by Blom [Blom 1985]. This scheme allows any pair of nodes in a network to find a pairwise key in a secure way as long as no more than  $\lambda$  nodes are compromised. The scheme is built on two matrices over a finite field  $GF(q)$ : a publicly known matrix  $G$  of size  $(\lambda + 1) \times N$ ; a secret matrix  $D$  of size  $(\lambda + 1) \times (\lambda + 1)$  created by key distribution center. The matrix  $A$  of size  $N \times (\lambda + 1)$  is then created as  $A = (D \cdot G)^T$  over the finite field  $GF(q)$ . Each row of  $A$  is the keys distributed to a group member and the row number can serve as a sensor's *id*. Since  $K = A \cdot G$  is a symmetric matrix, nodes  $i$  and  $j$  can generate a shared key ( $K_{ij}$  or  $K_{ji}$ ) from their predistributed secrets, where  $K_{ij}$  is the element in  $K$  located in the  $i$ th row and  $j$ th column.

A key pool is constructed by many key spaces, represented by  $A^{(t)}$ , where  $t = 1, \dots, \omega$ . Each sensor randomly selects  $\tau$  key spaces out of  $\omega$  key spaces, where  $\tau < \omega$ . If sensor  $k$  selects key space  $A^{(t)}$ , the  $k$ th row of  $A^{(t)}$  and  $k$ th column of  $G$  are preinstalled in the sensor (note that the  $G$  matrix is unique). The SK-RKP scheme has following properties:

- Once two nodes  $i$  and  $j$  have keys preinstalled from the same key space  $A^{(t)}$ , they can derive a shared key  $K_{ij}^{(t)} = K_{ji}^{(t)}$ .
- If  $x$  rows of a key space  $A^{(t)}$  are predistributed to  $x$  sensors and  $x \leq \lambda$ , any subset of the  $x$  sensors cannot collude to derive the secrets in other sensors.

—The *id* of a sensor is represented by the row number of the key matrix  $A$ . No other sensor can impersonate this sensor, since the row of  $A$  is uniquely distributed to this sensor.

The technical details refer to [Liu and Ning 2003a; Liu et al. 2005; Du et al. 2003; Du et al. 2005].

## B. MULTI-PATH ALGORITHMS

### ALGORITHM B.1 ENCODING OF RS CODES.

1. let  $m(x) = m_0 + m_1x + \dots + m_{k-1}x^{k-1}$  be the message polynomial to be encoded, where  $m_i \in GF(2^q)$  and  $k = n - 2t$ .  
 $g(x) = (x+\alpha)(x+\alpha^2) \dots (x+\alpha^{2t})$  is a publicly known generator polynomial, where  $\alpha^i \in GF(2^q)$  and  $g(x)$  has  $\alpha, \alpha^2, \dots, \alpha^{2t}$  as roots.
2. Dividing  $x^{2t}m(x)$  by  $g(x)$ , we have  $x^{2t}m(x) = a(x)g(x) + b(x)$ , where  $b(x) = b_0 + b_1x + \dots + b_{2t-1}x^{2t-1}$  and  $b(x)$  is the parity check polynomial.  
 Then  $c(x) = b(x) + x^{2t}$  is the codeword polynomial for the message  $m(x)$

### ALGORITHM B.2 DECODING OF RS CODES.

1.  $c(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1}$   
 $r(x) = r_0 + r_1x + \dots + r_{n-1}x^{n-1}$   
 $e(x) = e_0 + e_1x + \dots + e_{n-1}x^{n-1}$   
 where  $c_i, r_i, e_i \in GF(2^q)$ ,  $r(x)$  is received codeword, and  $e(x) = r(x) - c(x)$  is the error polynomial, where  $e_i = r_i - c_i$  is a symbol in  $GF(2^q)$ .
2. Suppose  $e(x)$  has  $v$  errors at the locations; then,  $e(x) = e_{j_1}x^{j_1} + e_{j_2}x^{j_2} + \dots + e_{j_v}x^{j_v}$ .  
 Faulty path can be identified by evaluating  $S_i = r(x)|_{x=\alpha^i} = r(\alpha^i)$ ,  $i = 1, \dots, 2t$ , if  $S_i \neq 0$ .  
 The error-location numbers are  $X_{j_1} = \alpha^{j_1}, X_{j_2} = \alpha^{j_2}, \dots, X_{j_v} = \alpha^{j_v}$   
 Using Forney's algorithm [Wicker 1995], the error values are  $Y_l = e_{j_l}$ ,  $l = 1, \dots, v$ .
3. To recover the the original message  $c(x)$ , we have  $c(x) = r(x) - e(x)$ .

## C. SENSOR COVERAGE – WITHIN THE SAME ZONE

The covering area of sensor  $[(i, j), b]$  in its zone  $Z(i, j)$  is shown in Figure 10(a). We can further divide the zone into 4 areas. The sensor coverage in these 4 areas are horizontally and vertically mapped to each other. Within a small area, there are two scenarios that shown in the Figure 10(a).

In the first scenario (shown the sensor is located at the position  $(x_1, y_1)$ ), the distance between the origin and the sensor is  $\sqrt{x_1^2 + y_1^2} \leq R$ . The coverage is composed by a sector with the angle  $\theta_1$  plus two triangles (the shaded areas). The  $\theta_1 = \frac{3}{2}\pi - \sin^{-1} \frac{B_1 - x_1}{R} - \sin^{-1} \frac{A_1 - y_1}{R}$ , where,  $A_1 - y_1 = \sqrt{R^2 - x_1^2}$ , and  $B_1 - x_1 = \sqrt{R^2 - y_1^2}$ .

The coverage (the shade area) of sensor  $[(i, j), b]$  in zone  $Z(i, j)$  is represented as  $C_b^1(i, j)$  and it is computed as follows:

$$C_b^1(i, j)|_{(x_1, y_1)} = x_1y_1 + \frac{1}{2} [x_1(A_1 - y_1) + y_1(B_1 - x_1)] + \frac{\theta_1}{2} R^2.$$

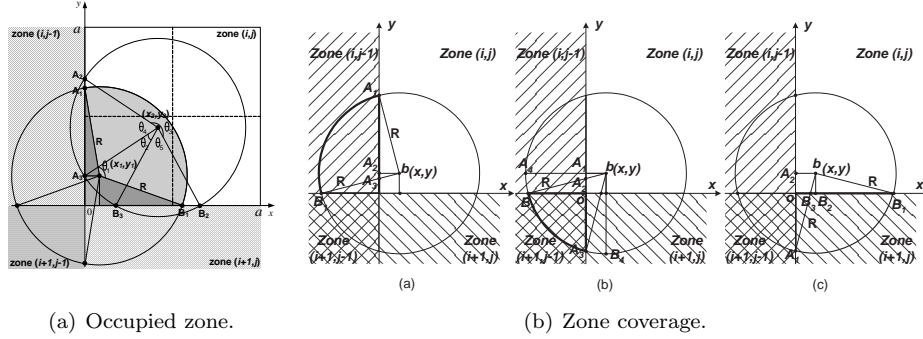


Fig. 10. Zone coverage among different zones.

In the second scenario (shown the sensor is located at the position  $(x_2, y_2)$ ), the distance between the origin and the sensor  $\sqrt{x_2^2 + y_2^2} > R$ . The coverage is composed by two triangles ( $\triangle A_2A_3o, \triangle B_2B_3o$ ) and two sectors with angles  $\theta_2$  and  $\theta_3$ . The  $\theta_2$  is given as  $\theta_2 = \frac{\pi}{2} - \sin^{-1} \frac{x_2 - B_3}{R} - \sin^{-1} \frac{y_2 - A_3}{R}$ . The  $\theta_3$  is given as  $\theta_3 = \frac{3}{2}\pi - \sin^{-1} \frac{B_2 - x_2}{R} - \sin^{-1} \frac{A_2 - y_2}{R}$ . The  $\theta_4$  and  $\theta_5$  are given as  $\theta_4 = 2 \sin^{-1} \frac{A_2 - y_2}{R}$  and  $\theta_5 = 2 \sin^{-1} \frac{B_2 - x_2}{R}$ , respectively. We note that  $y_2 - A_3 = A_2 - y_2 = \sqrt{R^2 - x_2^2}$  and  $x_2 - B_3 = B_2 - x_2 = \sqrt{R^2 - y_2^2}$ . The coverage (the shade area) of sensor  $[(i, j), b]$  in zone  $(i, j)$  is represented as  $C_b^2(i, j)$  and it is computed as follows:

$$C_b^2(i, j)|_{(x_2, y_2)} = \begin{cases} \frac{1}{2}[x_2(A_2 - A_3) + y_2(B_2 - B_3)] + \frac{\theta_2 + \theta_3}{2} R^2 & , x \leq R, y \leq R \\ \frac{1}{2}x_2(A_2 - A_3) + \frac{2\pi - \theta_4}{2} R^2 & , x \leq R, y > R \\ \frac{1}{2}y_2(B_2 - B_3) + \frac{2\pi - \theta_5}{2} R^2 & , x > R, y \leq R \\ \pi R^2 & , x > R, y > R. \end{cases}$$

#### D. SENSOR COVERAGE – IN DIFFERENT ZONES

##### D.1 Zone Coverage $\mathcal{C}(i, j^-)$

Shown in Figure 10(b)-(a), we first compute the area of the triangle  $\triangle A_3A_1b = \frac{1}{2}(|A_1A_2| + |A_2A_3|)|A_2b|$ , where  $|A_1A_2| = \sqrt{R^2 - x^2}$ ,  $|A_3b| = \frac{x}{\cos(\angle A_3bA_2)} = \frac{xR}{\sqrt{R^2 - y^2}}$ ,  $|A_2A_3| = \frac{xy}{\sqrt{R^2 - y^2}}$ , and  $|A_2b| = \sqrt{|A_3b|^2 - |A_2A_3|^2}$ .

The area of triangle  $\triangle A_3oB_1 = \frac{1}{2}|B_1o||A_3o|$ , where  $|B_1A_3| = R - |A_3b|$ ,  $|A_3o| = y - |A_2A_3|$ , and  $|B_1o| = \sqrt{|B_1A_3|^2 - |A_3o|^2}$ .

The area of sector  $\widehat{B_1bA_1} = \frac{\angle A_1bB_1}{2} R^2$ , where  $\angle A_2bA_3 = \sin^{-1} \left( \frac{y}{R} \right)$ ,  $\angle A_2A_1b = \sin^{-1} \left( \frac{x}{R} \right)$ ,  $\angle A_1bA_2 = \frac{\pi}{2} - \angle A_2A_1b$ , and  $\angle A_1bB_1 = \angle A_1bA_2 + \angle A_2bA_3$ .

Thus, the shade area  $\mathcal{C}(i, j^-)$  is given as:

$$\begin{aligned} \mathcal{C}_b(i, j^-)|_{(x,y)} &= \widehat{B_1bA_1} + \triangle B_1A_3o - \triangle A_3A_1b \\ &= \frac{R^2}{2} \left[ \frac{\pi}{2} + \sin^{-1} \left( \frac{y}{R} \right) - \sin^{-1} \left( \frac{x}{R} \right) \right] - \frac{x}{2} \left( \sqrt{R^2 - x^2} + \frac{xy}{\sqrt{R^2 - y^2}} \right) \\ &\quad + \frac{1}{2} \left( y - \frac{xy}{\sqrt{R^2 - y^2}} \right) \sqrt{\left( R - \frac{xR}{\sqrt{R^2 - y^2}} \right) - \left( y - \frac{xy}{\sqrt{R^2 - y^2}} \right)}. \end{aligned}$$

### D.2 Zone Coverage $\mathcal{C}(i^+, j^-)$

Shown in Figure 10(b)-(b), we first compute the area of sector  $\widehat{B_4bA_4} = \frac{\pi R^2}{4}$ . Then, we can derive the following relations: the area of sector  $\widehat{B_1bA_4} = \frac{\sin^{-1}(\frac{y}{R})}{2} R^2$ ; the area of triangle  $\triangle bB_1B_3 = \frac{1}{2} \sqrt{R^2 - y^2} y$ ; the area of sector  $\widehat{B_4bA_3} = \frac{\sin^{-1}(\frac{x}{R})}{2} R^2$ ; the area of triangle  $\triangle bA_1A_3 = \frac{1}{2} \sqrt{R^2 - x^2} x$ . The shade area  $\mathcal{C}(i^+, j^-)$  is given as:

$$\begin{aligned} \mathcal{C}_b(i^+, j^-)|_{(x,y)} &= \widehat{B_4bA_4} - \widehat{B_1bA_4} - \triangle bB_1B_3 - \widehat{B_4bA_3} - \triangle bA_1A_3 + xy \\ &= \frac{\pi R^2}{4} - \frac{R^2}{2} \left[ \sin^{-1} \left( \frac{y}{R} \right) + \sin^{-1} \left( \frac{x}{R} \right) \right] - \frac{1}{2} \left( x \sqrt{R^2 - x^2} + y \sqrt{R^2 - y^2} \right) + xy. \end{aligned}$$

### D.3 Zone Coverage $\mathcal{C}(i^+, j)$

Shown in Figure 10(b)-(c), we first compute the area of triangle  $\triangle B_3B_1b = \frac{1}{2} (|B_1B_2| + |B_2B_3|) |B_2b|$ , where  $|B_1B_2| = \sqrt{R^2 - y^2}$ ,  $|B_3b| = \frac{y}{\cos(\angle B_3bB_2)} = \frac{yR}{\sqrt{R^2 - x^2}}$ ,  $|B_2B_3| = \frac{xy}{\sqrt{R^2 - x^2}}$ , and  $|B_2b| = \sqrt{|B_3b|^2 - |B_2B_3|^2}$ .

The area of triangle  $\triangle B_3oA_1 = \frac{1}{2} |A_1o| |B_3o|$ , where  $|A_1B_3| = R - |B_3b|$ ,  $|B_3o| = x - |B_2B_3|$ , and  $|A_1o| = \sqrt{|A_1B_3|^2 - |B_3o|^2}$ .

The area of sector  $\widehat{A_1bB_1} = \frac{\angle B_1bA_1}{2\pi} \pi R^2 = \frac{\angle B_1bA_1}{2} R^2$ , where  $\angle B_2bB_3 = \sin^{-1} \left( \frac{x}{R} \right)$ ,  $\angle B_2B_1b = \sin^{-1} \left( \frac{y}{R} \right)$ ,  $\angle B_1bB_2 = \frac{\pi}{2} - \angle B_2B_1b$ , and  $\angle B_1bA_1 = \angle B_1bB_2 + \angle B_2bB_3$ .

Thus, the shade area  $\mathcal{C}(i^+, j)$  is given as follows:

$$\begin{aligned} \mathcal{C}_b(i^+, j)|_{(x,y)} &= \widehat{A_1bB_1} + \triangle A_1B_3o - \triangle B_3B_1b \\ &= \frac{R^2}{2} \left[ \frac{\pi}{2} + \sin^{-1} \left( \frac{x}{R} \right) - \sin^{-1} \left( \frac{y}{R} \right) \right] - \frac{y}{2} \left( \sqrt{R^2 - y^2} + \frac{xy}{\sqrt{R^2 - x^2}} \right) \\ &\quad + \frac{1}{2} \left( x - \frac{xy}{\sqrt{R^2 - x^2}} \right) \sqrt{\left( R - \frac{yR}{\sqrt{R^2 - x^2}} \right) - \left( x - \frac{xy}{\sqrt{R^2 - x^2}} \right)}. \end{aligned}$$