Location-aware Key Management Scheme for Wireless Sensor Networks

Dijiang Huang, Manish Mehta, Deep Medhi, Lein Harn
{dh7ee,mmmef7,dmedhi,harnl}@umkc.edu
University of Missouri-Kansas City

ABSTRACT

Sensor networks are composed of a large number of low power sensor devices. For secure communication among sensors, secret keys must be established between them. Recently, several pairwise key schemes have been proposed for large distributed sensor networks. These schemes randomly select a set of keys from a key pool and install the keys in the memory of each sensor. After deployment, the sensors can set up keys by using the preinstalled keys. Due to lack of tamper-resistant hardware, the sensor networks are vulnerable to node capture attacks. The information gained from captured nodes can be used to compromise communication among uncompromised sensors. Du et al. [1], Liu and Ning [2] proposed to use the known deployment information to reduce the memory requirements and mitigate the consequences of node capture attack. Our analysis shows that the assumption of random capture of sensors is too *weak*. An *intelligent* attacker can selectively capture sensors to get more information with less efforts. In addition to selective node capture attack, all recent proposals are vulnerable to node fabrication attack, in which an attacker can fabricate new sensors by manipulating the compromised secret keys and then deploy the fabricated sensors into the sensor system. To counter these attacks, we propose a grid-group scheme which uses known deployment information. Unlike the pairwise key scheme using deployment information proposed by Du et al., we uniformly deploy sensors in a large area; instead of randomly distributing keys from a large key pool to each sensor, we systematically distribute secret keys to each sensor from a structured key pool. Our performance analysis shows that our scheme requires less number of keys preinstalled for each sensor and is resilient to selective node capture attack and node fabrication attack.

Categories and Subject Descriptors

C.2.0 [**Computer-communication networks**]: General—*Security and protection*

SASN'04, October 25, 2004, Washington, DC, USA.

General Terms

Design, Security

Keywords

key management, sensor networks, probabilistic key sharing

1. INTRODUCTION

Sensor networks are composed of a large number of low-power sensor devices. Typically, these networks are installed to collect sensed data from sensors deployed in a large area. SmartDust [3] and WINS [4] are examples of sensor network projects. Within the networks, sensors communicate among themselves to exchange data and routing information. Since the sensor networks are usually deployed in unattended or even hostile environments (such as battle fields), the sensor networks are vulnerable to various kinds of active and passive attacks on the communication protocols. This demands secure communication among sensors.

We define a secure channel or link as as a channel through which two nodes can communicate with each other using a secret key. The secure channel is said to be compromised if an attacker can compromise the secret key. Since the low-power sensor devices have very limited computational power, the symmetric-key systems are preferred to establish secure channels. As specified in [5], the number of sensor nodes deployed in 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 value for every other sensor in the system. One naïve solution to use a common key between every pair of sensors can overcome the storage constraints, but it offers weak security. Since, if one node is compromised, the entire system is compromised. Recently, Random Key Predistribution (RKP) schemes have been proposed [6, 7, 8, 9, 10] for large-scale distributed sensor networks. These schemes randomly select a set of keys from a large key pool and install the keys in the memory of each sensor. After deployment, the sensors can set up keys by using the preinstalled keys. Since the RKP schemes require limited number of keys preinstalled in the sensors, a sensor may not share a key with all of its neighbors. In this case, a Pairwise Key Establishment (PKE) scheme is required to set up a shared key with every neighbor.

In current RKP schemes, the analyses of the security strength are done on the basis of number of communication links that can be compromised due to compromised sen-

sors in the network. In other words, the schemes consider probable use of the keys, exposed due to captured sensors, in non-compromised parts of the network. This attack is called node capture attack. Also, the current schemes consider random capture of nodes in the deployment region. To mitigate the random node capture attack, Du et al. [1] and Liu and Ning [2] proposed using deployment information (sensor location information) to improve the resilience to 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 which can provide more information for attackers to attack the sensor networks. In addition, due to lack of node authentication, attackers can easily fabricate nodes by using the secrets preinstalled in the captured node.

In this paper, we propose a new scheme, called Grid-group deployment scheme. This scheme utilizes merits from both [1] and [2]. Similar to [1, 2], a sensor deployment area is partitioned into multiple small square areas (zones) and the sensors deployed in each zone form a group. In the key predistribution phase, using the unconditionally secure and λ-collusion resistant properties of the group keying scheme proposed in [11], we utilize the key predistribution scheme proposed in [8, 9] to distribute keys for the sensors in each zone; for each sensor, we select a sensor in each of its adjacent zones and assign a unique key to them (the selection of the pair of sensors is based on the mapping between the unique node IDs assigned to the sensors; the technical details are presented in Section 4.2.2). After the deployment of sensors, each sensor first sets up pairwise keys with all its neighbors within its zone; then it sets up pairwise key with its neighbors located in adjacent zones. Comparing with previously proposed schemes, our approach is resilient to selective node capture attack and node fabrication attack. Our main contributions in this paper are as follows:

- We point out the weak assumption of random capture of nodes in current RKP schemes and introduce *selective attack* on RKP schemes. In particular, we show the importance of selective attack in RKP schemes that use deployment information.
- We point out the *node fabrication attack* on current RKP schemes and countermeasures for the same.
- We propose a new RKP scheme called Grid-group deployment scheme which is based on current schemes and deployment information. Further, this scheme is resilient against the introduced selective attack and node fabrication attack.
- Our proposed scheme reduces the number of keys preinstalled in each sensor.

The rest of the paper is organized as follows: In section 2, we provide the background of RKP schemes in sensor networks. We introduce attacks on current schemes in section 3. Section 4 describes our proposed scheme. The sensor area coverage analysis for the proposed scheme is given in section 5. In section 6, we analyze the *key graph* connectivity based on our proposed scheme. The Pairwise Key Establishment protocol is presented in section 7. A performance analysis addressing storage requirements, security, communication overhead, and computation overhead is given in section 8. Section 9 provides summary and future work.

2. BACKGROUND OF RANDOM KEY PREDISTRIBUTION SCHEMES

In this section, we review Purely Random Key Predistribution (P-RKP) schemes [6, 7] and Structured Key-pool Random Key Predistribution (SK-RKP) schemes [8, 9].

2.1 The Phases in Random Key Predistribution Schemes

We present the main phases for random key predistribution schemes [6, 7, 8, 9] as follows:

1. *Key predistribution phase*: A centralized key server generates a large key pool offline. The procedure for offline key distribution is as follows: 1. Assign a unique node identifier or key ring identifer to each sensor, 2. Select m different keys for each sensor from the key pool to form a key ring, 3. Load the key ring into the memory of the sensor.
2. *Sensor deployment phase*: The sensors are randomly picked and uniformly distributed in a large area. Typically, the number of neighbors of a sensor (n') is much smaller than the total number of deployed sensors (N).
3. *Key discovery phase*: During the key discovery phase, each sensor broadcasts its key identifiers in clear-text or uses private share-key discovery scheme[1] to discover the keys shared with its neighbors. By comparing the possessed keys, a sensor can build the list of reachable nodes with which share keys and then broadcast its list. Using the lists received from neighbors, a sensor can build a *key graph* (see Definition 1) based on the key-share relations among neighbors.
4. *Pairwise key establishment phase*: If a sensor shares key(s) with a given neighbor, the shared key(s) can be used as their pairwise key(s). If a sensor does not share key(s) with a given neighbor, the sensor uses the *key graph* built during key discovery phase to find a *key path* (see Definition 2) to set up the pairwise key.

The set of all neighbors of sensor i is represented by W_i. The definition of *key graph* is given as follows:

DEFINITION 1 (*key graph*). *A key graph maintained by node i is defined as $G_i = (V_i, E_i)$ where, the vertices set $V_i = \{j | j \in W_i \vee j = i\}$, the edges set $E_i = \{e_{jk} | j, k \in W_i \wedge j \mathcal{R} k\}$, $\mathcal{R}$ is a relation defined between any pair of nodes j and k if they share required number of key(s) after the key discovery phase.*

The definition of *key path* is given as follows:

DEFINITION 2 (*key path*). *A* key path *between node A and B is defined as a sequence of nodes A, N_1, N_2,..., N_i, B, such that, each pair of nodes (A, N_1), (N_1, N_2), ..., (N_{i-1}, N_i), (N_i, B) has required number of shared key(s) after the key discovery phase. The length of the key path is the number of pairs of nodes in it.*

[1]Specified in [6], using private share-key discovery, for every key on a key ring, each node could broadcast of list α, $E_{K_i}(\alpha), i = 1, \ldots, k$, where α is a challenge. The decryption of $E_{K_i}(\alpha)$ with the proper key by a recipient would reveal the challenge α and establish a shared key with the broadcasting node.

2.2 Purely Random Key Predistribution (P-RKP) Schemes

For current P-RKP schemes, the phases presented in Section 2.1 can be applied without any change. There are two characteristics of P-RKP schemes. First, the m keys preinstalled in a sensor can also be installed in other sensors. That is, a key can be shared by more than one pair of sensors. Second, in most of current schemes, there is no relation between the set of preloaded keys and the sensor *id*. A recent solution proposed by Pietro et al. [12] attempts to define this relation. However, the scheme is not scalable in that the size of the network is restricted by a function of number of preinstalled keys.

2.3 Structured Key-pool Random Key Predistribution (SK-RKP) Scheme

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. This link can prevent the attackers from misusing the sensors' *ids*. In the following paragraphs, we give a brief description of structured key pool scheme.

The SK-RKP scheme uses the key predistribution scheme proposed by Blom [13]. 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 λ nodes are compromised. 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 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 ith row and jth column.

A key pool is constructed by many key spaces, represented by $A^{(t)}$, where $t=1,\ldots,\omega$. Each sensor randomly selects τ key spaces out of ω key spaces, where $\tau<\omega$. If sensor k selects key space $A^{(t)}$, the kth row of $A^{(t)}$ and kth 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 presinstalled 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 [8, 9].

3. ATTACKS ON RANDOM KEY PREDISTRIBUTION SCHEMES

The proposed P-RKP schemes and SK-RKP schemes have several limitations which make them vulnerable to attacks. Since sensors are low-cost devices and operate in unattended environment for many applications, they cannot be considered tamper-resistant. We make following assumptions about capabilities of the attacker.

- 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 and record all the traffic in the network.
- The attacker has ability to physically locate a given sensor by listening to the traffic.
- The attacker has ability to fabricate similar nodes and deploy them.

3.1 Selective Node Capture Attack

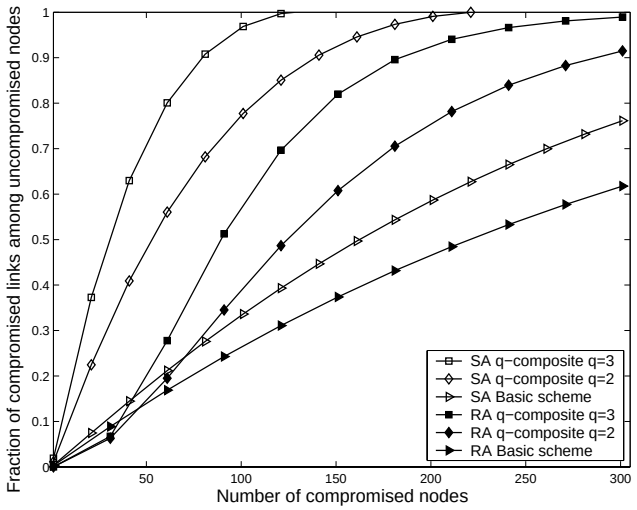
In all current RKP proposals, the sensors are assumed to be captured randomly. But in practice, the random capture assumption is too weak. The attacker can purposely attack certain area or a group of sensors possibly located closer to each other. Thus, an attacker can purposely locate a sensor and compromise the sensors which can give him more information about the sensor network. For example, in P-RKP scheme, each sensor broadcasts its key ring *id* (the key list). An attacker can selectively compromise a sensor that possesses the most number of keys that are not already compromised.

We model the selective attack by using the heuristic technique. In the following presentations, x is the number of compromised sensors, C_x is the cardinality of the set of compromised keys when x nodes are compromised, P is the size of the key pool, m is the number of keys preinstalled in each sensor, N is the total number of sensors deployed in the network, and k is a variable. We use B to represent the threshold that an attacker is to inspect and then to decide which sensors to capture next. The mathematical model of selective attack is presented as follows:

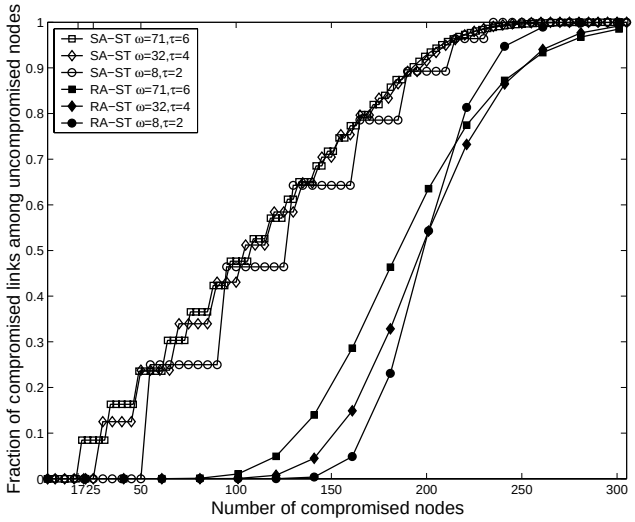
$$B = \frac{\binom{P-C_x}{m-k}\binom{C_x}{k}}{\binom{P}{m}}(N-x) \qquad (k=0,\ldots,m) \quad (1)$$

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

The heuristic method is described as follows. Initially, when $k=0$, an attacker can arbitrarily capture a sensor and derive m keys preinstalled in the captured sensor and $C_x=m$. Then, he inspects the B: if $B\geq 1$, he continuously captures the nodes with $m-k$ keys that are not already compromised, and for each capture, C_x is increased by $m-k$; if $B<1$, he increases k by 1 until $B\geq 1$. He then captures the sensors with $m-k$ keys that are not already compromised. The attacker continues this process until the



(a) Selective attack on P-RKP schemes ($m = 100$, $p_1 = 0.423$, N=10000; basic scheme $P = 28140$; q-composite scheme, when $q = 2$, $P = 9120$, when $q = 3$, $P = 5220$)



(b) Selective attack on SK-RKP schemes ($m = 100$, $p_1 = 0.423$, N=10000)

Figure 1: Selective attack on RKP schemes.

condition $m = k$ is fulfilled or the entire key pool is compromised. The condition $B > 1$ means there exists uncompromised sensor that has $m - k$ keys that are not already compromised. Figure 1(a) shows the comparison between the *selective-node-capture attack* (SA) and *random-node-capture attack* (RA) on the P-RKP schemes. It might be note that, in this figure, the size of key pool P is computed based on the *key graph* connectivity probability presented in [14] and p_1 is the probability that two sensors share at least one key during the key discovery phase. Our studies show that the selective node capture attack can gain more information than random node capture attack with the same number of captured sensors.

In SK-RKP scheme, the attacker can selectively capture the sensors that possess keys within the same key space. Once $\lambda+1$ sensors with preinstalled keys from the same key space are compromised, all the keys allocated from the same key space are compromised. Thus, an attacker can incrementally compromise the sensors that use same key space. In this way, the attacker can compromise all the key spaces one-by-one. Since sensors have keys from more than one key space preinstalled, the number of sensors required to be captured to compromise the subsequent key spaces is smaller. Figure 1(b) shows the SK-RKP scheme [8, 9] under SA and RA, with $m = 100$, $p_1 = 0.432$. This figure shows that under selective attack, the robustness (threshold) of SK-RKP scheme against node capture attack decreases dramatically. In the example, the threshold values under SA are: 17 with $\omega = 71, \tau = 6$; 25 with $\omega = 32, \tau = 4$; 50 with $\omega = 8, \tau = 2$. The relation between the τ and the threshold is: the smaller the τ, the higher the initial threshold. That is, with $\tau = 2$, we can maximize the initial threshold.

3.2 Active Attack: Node Fabrication Attack

The proposed P-RKP and SK-RKP schemes are all vulnerable to node fabrication attack. We describe the node fabrication attack as follows: in this attack, the attacker compromises only few sensors and uses the captured keys to fabricate sensors with identities of uncompromised sensors or fabricate sensors with new identities. Then, the attacker can deploy the fabricated nodes in the parts of the network where the original node is not present. The uncompromised sensors in the network cannot detect the fabricated nodes as anomalous nodes as long as they can have standard communication with them. This attack is severer as compared to passive listening attacks as the attacker may have enough information to fabricate many sensors with many different identities and possibly outnumber the original set of sensors.

The attacker can launch the node fabrication attack on P-RKP schemes [6, 7] by capturing only two sensors. Since there is no *id* authentication by using P-RKP scheme, 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.

SK-RKP scheme is also vulnerable to node fabrication attack. However, there are few restrictions for the attacker. First, an attack requires to capture more than λ sensors in order to compromise a key space. Second, an attacker cannot arbitrarily generate new *ids* for the fabricated sensors, since the *ids* indicate the rows of the secret matrix A possessed by the sensors. A wrong *id* will not guarantee that a fabricated sensor can set up a pairwise key with uncompromised sensors. Thus, by restricting the distribution of the number of rows of a secret matrix A to λ, we can prevent the node fabrication attack. Our scheme described in the next section uses this technique. The previous proposals [8, 9] cannot fulfil this requirement with relatively small λ to support a large sensor system with typically 1,000 to 10,000 sensors. Since in our scheme, we restrict the size of deployment region by partitioning it into multiple zones, the required network size for each zone becomes smaller and we can prevent node fabrication attacks by using the technique

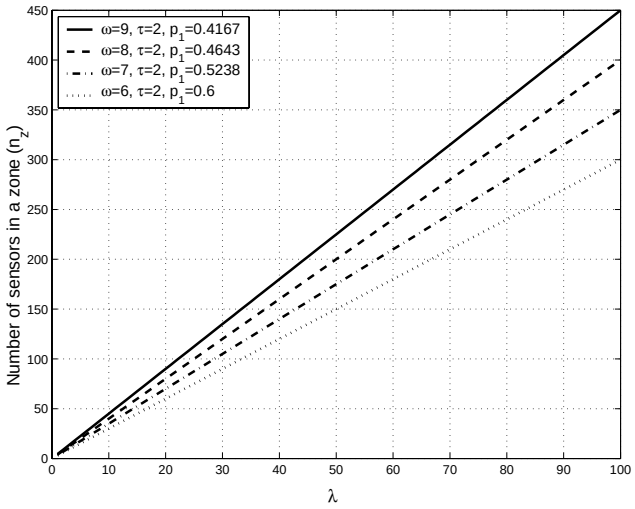


Figure 2: Relation between λ and the number of sensors deployed in a zone (n_z), and $m = 100$

discussed above. Figure 2 shows the relation between the value of λ and supported size of network for each zone (n_z).

4. GRID-GROUP DEPLOYMENT SCHEME

In this section, we present our grid-group deployment scheme and the key predistribution schemes used for the same.

For our scheme, we assume that the target deployment area is a two-dimensional rectangular region with the size $(i \cdot a) \times (j \cdot a)$ square meters. The rectangular region can be further divided into $(i \cdot j)$ deployment areas, each of size $a \times a$ square meters. In this paper, we denote each small deployment area as *zone* $\mathcal{Z}(i, j)$, where $Area(\mathcal{Z}(i, j)) = a^2$. An example of deployment region is shown in Figure 3, where $i = j = 6$. We use $G(i, j)$ to denote the group of sensors deployed in zone $\mathcal{Z}(i, j)$. We assume that the sensors are uniformly distributed over the deployment region and for each group, the number of sensors in the group is n_z. We denote the total number of sensors in the whole deployment region by N. Thus, we have $N = n_z \cdot i \cdot j$. A sensor is identified by $[(i, j), b]$, where (i, j) is the group *id*, and b is the unique node *id* of a sensor ($b = 1, \ldots, N$).

4.1 Sensor Deployment Method

The sensor deployment method is given as follows:

- Partition N sensors into $i \cdot j$ groups with n_z sensors in each group.
- Assign the identifier $[(i, j), b]$ to each sensor in the $G(i, j)$, where $b = 1, \ldots N$.
- Assign m keys to each sensor in group $G(i, j)$.[2]
- Uniformly distribute the sensors for the group $G(i, j)$ in zone $\mathcal{Z}(i, j)$.

[2]The key-assignment is presented in Section 4.2.

4.2 Key Predistribution Schemes

The value (i, j) is used to identify a group or zone. We use the superscripts $^+$ and $^-$ with i and j to denote the neighboring groups or zones. For example, $G(i^+, j) = G(i+1, j)$ is a neighboring group of $G(i, j)$. We propose two key predistribution schemes according to the group relations. 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*.

4.2.1 I-Scheme: key predistribution within a given zone

We use the scheme specified in Section 2.3 for our *I-Scheme*. To achieve the *non-colluding*[3] property. We set the following restrictions:

- $\tau = 2$, this maximizes the initial node capture threshold (refer to Figure 1(b)).
- No more than λ sensors are allowed to choose a given key space.
- As a result of the first two restrictions, we restrict the number of sensors in each group, n_z, to $|G(i, j)| \leq \lambda\omega/\tau$.

As shown in Figure 1(b), for a fixed value of m=100, the number of keys preinstalled in each sensor, the smaller the τ, the higher the initial node capture threshold. The initial threshold is computed as m/τ. For example, for $\tau = 2$, $\omega = 8$, and $m = 100$, the initial threshold is 50. Since $\tau = 2$ gives the highest initial threshold. In this paper, we consider $\tau = 2$ for our analysis.

For each key space, the secret matrix $A = (D \cdot G)^T$ is a $N \times (\lambda + 1)$ matrix. If an attacker has knowledge of more than λ rows, the entire matrix A can be derived. Thus, to improve the survivability of the sensor system, we restrict the number of rows of matrix A distributed to sensors to λ.

As a consequence of the previous restrictions, the number of sensors deployed in each zone is restricted to $|G(i, j)| \leq \lambda\omega/\tau$. Figure 2 shows the relation between $|G(i, j)|$ and λ. Based on above discussion, we propose the following key predistribution scheme (*I-Scheme*) for the sensors located within the same zone as follows:

1. The key pool $\mathcal{P}$ is composed by $P = L \times M$ sub-key pools (a sub-key pool is represented as $\mathcal{P}(i, j)$ where $i = 1, \ldots, L$, $j = 1, \ldots, M$). Each sub-key pool is divided into ω sub-key spaces. A sub-key space is a $N \times (\lambda + 1)$ key matrix A. Each element of A is a unique key.
2. Divide the N sensors in $L \times M$ groups (a group is represented by $G(i, j)$, where $i = 1, \ldots, L$, $j = 1, \ldots, M$).
3. Assign unique identifiers to the sensors. For each sensor, assign the $id = [(i, j), b]$, where (i, j) is the group *id*, and $b = 1, \ldots, N$.

[3]The *non-colluding* feature of the proposed pairwise key scheme is described as follows: for all pairwise key $K_{ij} \in \mathcal{P}$, where K_{ij} is the pairwise key that can be derived from the secrets possessed by sensors $s_i, s_j \in \mathcal{S}$, all sensors in the set $\mathcal{S} \setminus \{s_i, s_j\}$ cannot derive the pairwise key K_{ij}, $\mathcal{S}$ is the set of all sensors deployed in the system.

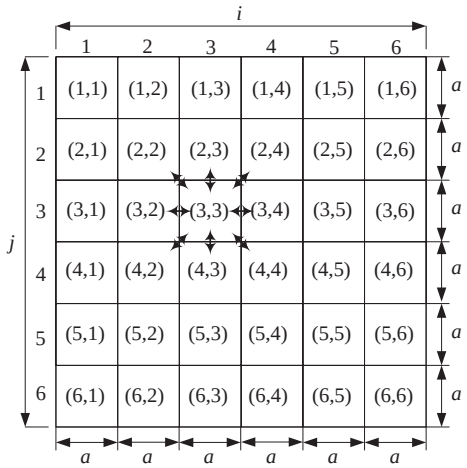


Figure 3: Sensor deployment in a grid structure

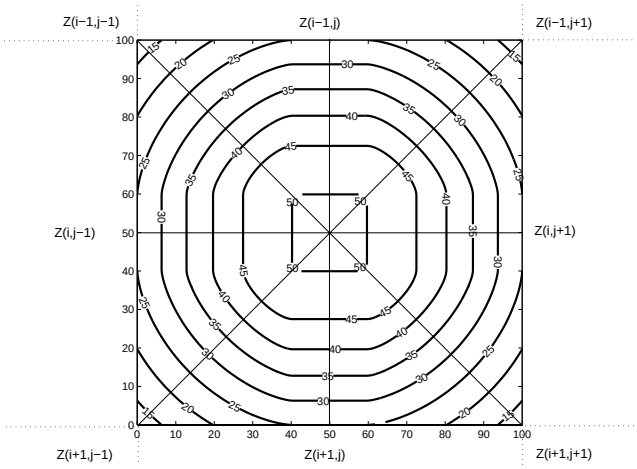


Figure 4: Contour curves for average number of neighbors within the same zone ($n' = 50$, $n_z = 100$)

4. For sensor $[(i,j),b]$, randomly select τ sub-key spaces from ω sub-key spaces in $\mathcal{P}(i,j)$ while making sure that the selected sub-key space is not already selected λ times. Load the sensor with the b^{th} row of matrix A for each sub-key space selected.

4.2.2 E-Scheme: keys predistribution for two adjacent zones

As shown in Figure 3, a zone can have the maximum of 8 neighboring zones; e.g., the bidirectional arrows shown around zone $\mathcal{Z}(3,3)$. Our key predistribution scheme (*E-Scheme*) for sensors in two adjacent zones is given as follows:

1. For a sensor i in group $G(i_1,j_1)$, randomly select one sensor, say j, 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 $|j_1-j_2|\leq 1$.

2. Install duple $< k_{ij}, id_j >$ in i and duple $< k_{ij}, id_i >$ in j, where key k_{ij} is unique and id_i, id_j are the identifiers of node i, j respectively. Once node i select a peer node j in group $G(i_2,j_2)$, it cannot select another node in the same group.

3. If all sensors have selected a node in each of its neighboring groups, stop; otherwise goto step 1.

5. AREA COVERAGE ANALYSIS

In this section, we study the area coverage of a sensor within the same zone and in the neighboring zones. Our analysis is based on the following assumptions:

- All our analysis is based on 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 $R \leq a/2$. In our analysis, we assume $R = 40$ meters, $a = 100$ meters
- 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 $n_z = a^2\rho \approx \left\lceil \frac{a^2 n'}{\pi R^2} \right\rceil$.

5.1 Sensor Coverage – within the Same Zone

We present the coverage of sensor $[(i,j),b]$ in its zone $\mathcal{Z}(i,j)$ in this section. 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 A. From the above results, the number of neighbors of sensor $[(i,j),b]$ within the zone $\mathcal{Z}(i,j)$ is given as:

$$\mathcal{N}_b(i,j) = \rho \cdot \mathcal{C}_b(i,j) \tag{2}$$

Where $\mathcal{N}_b(i,j)$ is the number of neighbors of sensor $[(i,j),b]$ within the zone $\mathcal{Z}(i,j)$. In Figure 4, we show the contour curves of the average number of neighbors of sensor $[(i,j),b]$ within the zone $\mathcal{Z}(i,j)$.

5.2 Sensor Coverage – in Different Zone

In Figure 4, we show that there are 8 possible zones surrounding zone $\mathcal{Z}(i,j)$. 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 $\mathcal{Z}(i+1,j-1)$. 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 $\mathcal{Z}(i,j-1)$.

The representations and proofs of neighboring zone coverage $\mathcal{C}_b(i,j^-)|_{(x,y)}$, $\mathcal{C}_b(i^+,j^-)|_{(x,y)}$, and $\mathcal{C}_b(i^+,j)|_{(x,y)}$ are given in Appendix B. In summary, the number of neighbors that node $[(i,j),b]$ covers in a neighboring zone is given as:

$$\mathcal{N}_b(i^*,j^*) = \rho \cdot \mathcal{C}_b(i^*,j^*) \tag{3}$$

where * represents -, +, or none.

6. KEY GRAPH CONNECTIVITY

In this section, we present the *Key Graph* connectivity analysis for sensors located within the same zone and in adjacent zones.

6.1 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 3, we select a unique key pool for each zone, i.e., $\mathcal{P}(i,j)$ for $\mathcal{Z}(i,j)$.

To determine the size of key pool, $|\mathcal{P}(i,j)|$, and the number of keys selected, m, from the key pool for sensor $[(i,j),b]$, we use the equations of P-RKP scheme proposed by Eschenauer and Gligor [6] and further modified for SK-RKP scheme by Du et al. [9].

$$p_1 = 1 - \frac{\binom{\omega}{\tau}\binom{\omega-\tau}{\tau}}{\binom{\omega}{\tau}^2} = 1 - \frac{((\omega-\tau)!)^2}{(\omega-2\tau)!\omega!} \tag{4}$$

Here, p_1 is the probability that given two sensors share at least one key. Eschenauer and Gligor [6] proposed an approximate method to compute *key graph* connectivity. Our preliminary studies show that this approach does not work well when the neighborhood size of a sensor is small. We derive the *key graph* connectivity by using binomial probability distribution and modified binomial probability distribution in a heuristic hop-by-hop fashion. The *key graph* connectivity probability is presented in [14].

Since we assume that the sensors are uniformly distributed within a zone, the closer the sensor to the center of the zone, the more the neighbors within the same zone for the sensor. Thus, the *key graph* connectivity will only be considered as the *key graph* created by the sensors within the same zone. For sensor $[(i,j),b]$ in zone $\mathcal{Z}(i,j)$, the number of neighbors within its zone is $C_b(i,j)$. As shown in Figure 4, 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 *key graph*[4] connectivity presented in [14], we derive the probability $p_1 = 0.5$. When $p_1 \geq 0.5$, the *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 *key graph* is connected within five hops is 0.8736 and on average there are only 0.1483 neighbors that can not be reached within five hops. If there exist unreachable nodes, a sensor can just simply send requests to is neighbors which have more than 25 neighbors to set up the pairwise keys. A neighbor with more than 25 neighbors can be identified from the neighbor-list broadcasted during the key discovery phase. Since we know that a sensor with 25 or more neighbors can set pairwise keys with all its neighbors, it can help to set up the pairwise key when they all within each other's communication range. In Section 7.1, we will discuss how a sensor sends requests to its neighbors for pairwise key set up within the same zone.

[4]Here, the key graph is composed by only the sensors within the same zone.

6.2 Key Graph Connectivity between Two Adjacent Zones

The node $[(i,j),b]$ may be located close to the boundary of two neighboring zones, $\mathcal{Z}(i,j)$ and $\mathcal{Z}(i^*,j^*)$. 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 $\mathcal{C}_b(i^*,j^*)$. The pairwise key establishment protocol between two adjacent zones is described in Section 7.2. Thus, using (2) and (3), 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.

$$p(i^*,j^*) = 1 - \frac{\binom{n_z - \mathcal{N}_b(i,j)}{\mathcal{N}_b(i^*,j^*)}\binom{n_z - \mathcal{N}_b(i^*,j^*)}{\mathcal{N}_b(i,j)}}{\binom{n_z}{\mathcal{N}_b(i^*,j^*)}\binom{n_z}{\mathcal{N}_b(i,j)}} \tag{5}$$

Note that (2) and (3) 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 (5), we draw the probability contour curves that a node in $\mathcal{Z}(i,j)$ can connect to its neighboring zones $\mathcal{Z}(i,j^-)$ and $\mathcal{Z}(i^+,j^-)$ with parameters $a = 100m, R = 40m, n' = 50, n_z = 100$ in Figure 5(a).

7. PAIRWISE KEY ESTABLISHMENT PROTOCOL

Our key establishment protocol[5] consists of two phases: 1. the key establishment within a given zone, 2. the key establishment between adjacent zones.

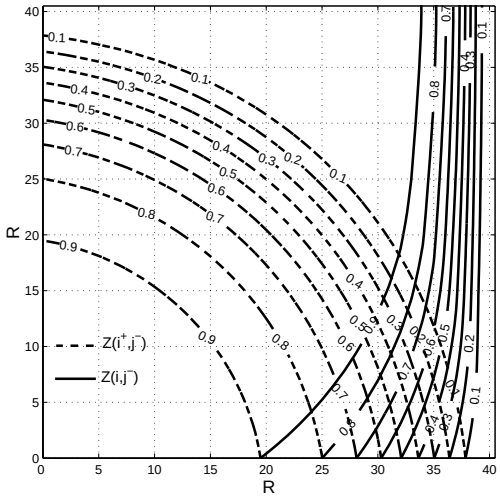
7.1 Key Establishment within the Same Zone

The key establishment within the same zone is the first phase after deployment of the sensors. In this phase, each sensor attempts to establish pairwise keys with all its neighbors within the same zone.

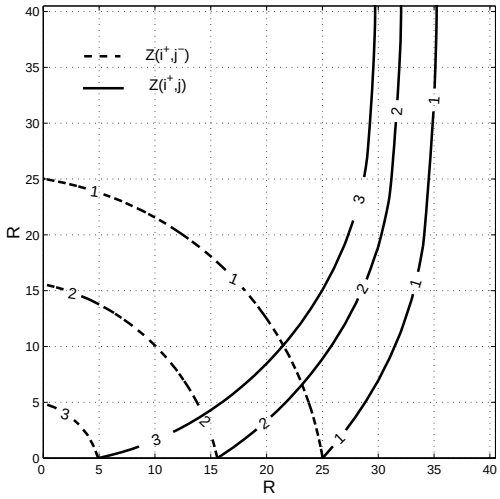
In our scheme, each sensor, say $[(i,j),b]$, initiates this phase by broadcasting its identifier $[(i,j),b]$ and its key space identifiers $[\tau_1,\tau_2]$. Based on the received *ids* and corresponding key spaces, a sensor builds a *key graph* with *ids* of all the neighbors as vertices. For each of the neighbors, say $[(i,j),u]$, the sensor checks if they share the same key space. If they do share a key space, they can derive the pairwise key K_{bu} using the key agreement method presented in Section 2.3. The node $[(i,j),b]$ will then add a link between itself and node $[(i,j),u]$ in its *key graph*.

After receiving the identifiers from all neighbors and adding links in the *key graph* for the neighbors with shared key space, the sensor broadcasts a list of neighbors who share key space with it. After receiving the same type of list from the neighbors, each sensor updates its *key graph* by adding edges between vertices according to the received neighbor-list. Finally, based on the derived *key graph*, the sensor can use *source routing*, by explicitly specifying the key path (in hop-by-hop fashion), to send request and establish pairwise keys with all its remaining neighbors.

[5]Due to page limits, in this paper, we only present an outline of the key establishment protocol. The detailed protocol will be covered in our following papers.



(a) Probability contour curves to zone $\mathcal{Z}(i^+, j^-)$ and zone $\mathcal{Z}(i, j^-)$ with q=1



(b) Connectivity contour curves to the neighboring zone $\mathcal{Z}(i^+, j^-)$ and zone $\mathcal{Z}(i, j^-)$ with q=1, 2, 3. $p_{\hat{q}}(i^*, j^*) = 0.8$

Figure 5: Connectivity between two adjacent zones ($a = 100m, R = 40m, n' = 50, n_z = 100$)

There may be a few nodes that may not be able to set up pairwise keys to all its neighbors within a given zone (e.g., the node are located outside of 25 curve line in Figure 4). In Section 6.1, we present the analysis that this probability is small. Even if there exist nodes that cannot set up pairwise keys with its neighbors, they can refer to the nodes who have already set up the pairwise keys with all its neighbors within the same zone. These nodes are usually located around the high numbered contour curves. For example, if our selected thresholds ω and τ can fulfil the requirement that a node with 25 or more neighbors can set pairwise keys with all its neighbors, the nodes locate around 25 curve line can set up links to every node within its communication range which includes the nodes located outside the 25 curve line. Now, these node can serve as the intermediate nodes (or proxy) to set up the pairwise keys. A node can send requests to its neighbors with more than 25 neighbors within the same zone. If the node has a link to the requested destination, it selects a pairwise key and encrypts it using the already set up pairwise keys with the source and destination. The source (a requestor) can send multiple requests to its neighbors and it may receive multiple responses. In this case, the source and destination node can exclusive-or all received keys and use the result as their pairwise key. This arrangement improves security. A neighbor with more than 25 neighbors can be identified from the neighbor-list broadcasting during the key discovery phase.

7.2 Key Establishment between Adjacent Zones

After the first phase of key establishment, a sensor sets up pairwise keys with all its neighbors within the same zone. Then, the system goes into the second phase of key establishment to set up pairwise keys with nodes located in the adjacent zones.

We described the key predistribution scheme for two adjacent zones (*E-Scheme*) in Section 4.2.2. When a sensor wants to set up keys 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 key with the nodes in the requestor's list acts as a proxy and does the following: 1. selects a pairwise key for the pair, 2. encrypts the selected pairwise key using the pairwise key already set up between itself and the requestor and the pairwise key already shared between itself and the destination node, 3. sends the two encrypted messages to the requestor. Upon receiving the response, the requestor will forward the encrypted pairwise key to the destination. Figure 5(a) shows the contour curves of the probabilities that a node in $\mathcal{Z}(i, j)$ can connect to its neighboring zones $\mathcal{Z}(i, j^-)$ and $\mathcal{Z}(i^+, j^-)$. Since during the first phase, nodes have already set up pairwise keys to all their neighbors within the same zone, during the second phase, as long as there exists one node with a link to the neighboring zone, it can be used as a bridge to set up pairwise keys to the neighboring zone for all its neighbors.

The probability that a node can connect to neighboring zones with the help of exactly k neighbors is given as follows:

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

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

$$p_{\hat{q}}(i^*, j^*) = 1 - [p_0(i^*, j^*) + \cdots + p_{q-1}(i^*, j^*)] \qquad (6)$$

Figure 5(b) 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 who 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. If there are q keys generated, the pairwise key is given as:

$$k = k_1 \oplus \cdots \oplus k_q \quad (7)$$

where $\oplus$ is the exclusive-or operator.

Comparing Figure 4 and Figure 5(b), 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.

Similar to the key establishment scheme within the same zone, for a sensor who cannot set up a key path to a neighboring zone, a sensor can send requests to its neighbors with $15 \sim 30$ neighbors within the same zone. If the nodes has a link to the requested destination, it selects a pairwise key and encrypts it using the already established pairwise keys with the source and destination. The pair of nodes that may want to set up a pairwise key can utilize the function presented in (7) to exclusive-or all received keys and use the result as their pairwise key.

8. PERFORMANCE ANALYSIS

In this section, we present the performance analysis based on storage, security, communication overhead, and computation overhead due to the proposed scheme. We also provide comparison studies of our scheme with the sensor deployment scheme proposed in [1].

8.1 Other Location-aware Schemes

Two schemes have been proposed in [2] by using known sensor location information. The first scheme is called *closest pairwise keys scheme* (CPKS). This scheme assumes the deployment point of each sensor is known in advance, which is too strict for implementation. The second scheme is called *location-based key predistribution* (LBKP). This scheme partitions a deployment area in multiple square areas (zones); using the same key predistribution schemes proposed in [11], each area is associated with two polynomials; a key server is responsible for predistributing keys based on known network topology and sensors' location points within a square area. Once a sensor's location point is determined, the key server installs a set of polynomial variables associated with the square area in the sensor. Using this scheme, within a square area, if $\lambda + 1$ or more sensors are captured, all communications between this area and its adjacent area are compromised. Our analysis in Section 3.1 shows this scheme is vulnerable to selective node capture attack. To overcome this problem, instead of using two polynomial settings for each zone, we use a unique structured key pool (proposed in [8, 9]) for each zone and restrict at most λ polynomial variables that are distributed from each polynomial. Another vulnerability of the scheme in [2] is that, if attackers capture more than λ sensors, they can fabricate nodes without being detected. We overcome this problem by assign a unique number (from 1 to N) to N deployed sensors. This number also identify the column number of G matrix. Since this number is unique for a sensor and no attacker can compromise more than λ rows of a key matrix A. Thus, the attacker cannot fabricate new sensors by installing new row elements to a fabricated sensor. The LBKP scheme also requires the location of each sensor is preknown, which is too restricted for implementation due to large number of sensors. In our grid-group deployment scheme, we assume a group of sensors (100 sensors in our performance analysis) are uniformly deployed within a given square area, which reduces the deployment complexity. Our scheme employs two stage pairwise key establishment scheme which uses two set of preinstalled keys. This method also reduces the number of keys preinstalled in each sensor.

In [1], Du et al. showed that using known deployment information, the performance of P-RKP schemes can be improved significantly. Especially, the proposed scheme reduces the consequences of random node capture attack. The main contribution of this scheme is to restrict the shared key information locally (within a small range, e.g., $100m \times 100m$). The P-RKP and SK-RKP schemes proposed in [6, 7, 10, 8, 9] assume that each sensor has an equal probability to have a given key installed in its memory. Instead of distributing the keys uniformly within a given sensor system, Du et al. [1] proposed to restrict the key distribution locally by geographically dividing the deployment region into $N \times M$ small areas. A small key pool is constructed for each small area acco'rding to its neighboring-area relation. The sensors located within each small area (which form a group) use the P-RKP scheme to predistribute the keys. They call this deployment method as group-based deployment model. However, the scheme proposed by Du et al. has several deficiencies. The sensor density of of an area is uneven in that the density around the center of group deployment point is much higher than that at the edge of the small area. The deployment pattern can be modeled by normal distribution. Thus, in order to make sure that a sensor deployed at the edge of an area has enough preinstalled keys to set up pairwise keys with all its neighbors, the group-based deployment model requires more sensors deployed around the group deployment point. In addition, the uneven distribution of sensors within a given area may be undesirable from the application and the security point view. For example, the sensed data may be unevenly distributed within the deployed area which can introduce additional complexity for the applications analyzing the sensed data; moreover, an attacker can capture more sensors around the group deployment point.

8.2 System Configuration

In order to compare with the P-RKP location scheme, we use the similar system configuration proposed in [1].

- The number of sensor nodes in the sensor network is 10,000.
- The deployment area is $1000m \times 1000m$.
- The area is divided into a grid of size 100, with each square (a zone) of size $100m \times 100m$.
- The number of sensors deployed within each zone is $n_z = 100$.
- The communication radius R is 40m and the average number of neighbors of a sensor is $n' = 50$.
- The parameters used for key predistribution scheme within a zone are $\tau = 2$ and $\omega = 7$. Therefore, the probability that two neighboring sensors share a key is $p_1 = 0.5238$.

8.3 Storage Overhead Analysis

A sensor is required to store $m = (\lambda + 1)\tau$ keys that are used to set up pairwise key within its zone, where λ 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 key for 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 give as:

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

where the γ is the number of neighboring zones and α 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 [6] which requires $m = 272$ to fulfil $p_1 = 0.5238$, our scheme requires $m = 68$ which is much lesser. For the scheme specified in [1], to achieve the $p_1 = 0.5238$, it requires 72 keys preinstalled for each sensor, which is a marginally higher than our scheme.

8.4 Security Analysis

Our security analysis presents a new way to analyze the security of pairwise key establishment for large distributed sensor system. Particularly, selective attack and node fabrication attack have not been addressed thoroughly in current literatures. All our analysis is based on the attacker's capabilities presented in Section 3 and the two phases of key establishment procedure presented in Section 7.

8.4.1 Security evaluation metrics

Sensor networks have many characteristics that make them more vulnerable to attacks as compared to conventional computing environment. We present several criteria that represent desirable characteristics in a pairwise key establishment scheme for sensor networks.

Resilience against node capture attack: to evaluate the random node capture attack and selective node capture attack: we evaluate the fraction of compromised links among uncompromised nodes due to captured nodes.

Resilience against node fabrication: we evaluate a the resilience of scheme against node fabrication by evaluating the capability of the attacker to successfully deploy the fabricated sensors into the deployment area.

8.4.2 Random node capture attack

Random node capture attack assumes that an attacker randomly captures the deployed sensors. If the attacker can gain the information which is not already known to him, the attack is considered successful. To evaluate the resilience to against the random node capture attack, we use the fraction of compromised communication links among uncompromised nodes to evaluate the proposed schemes.

Our pairwise key establishment scheme includes two phases. The first phase uses the SK-RKP scheme proposed in [8, 9]. In addition, we restrict the number of rows of secret matrix A ($A = (D \cdot G)^T$) distributed to sensors to λ. The side effect of this restriction is that the number of sensors deployed in each zone is restricted by $\lambda\omega/\tau$, where τ is the number of key spaces selected for each sensor. This restriction gains the maximum security to guard against node capture attack

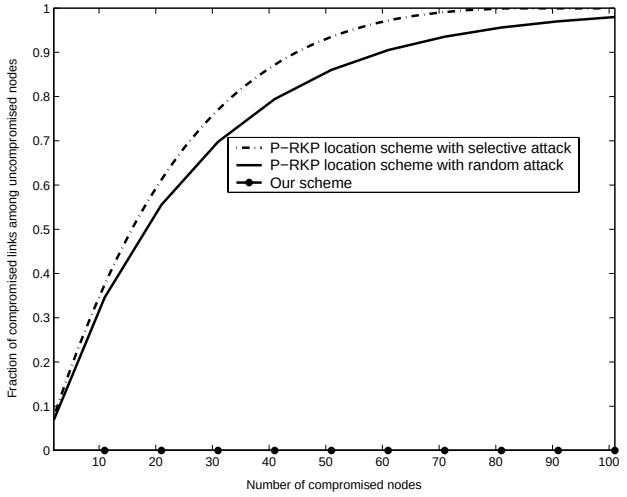


Figure 6: Random/Selective node capture attack: P-RKP location scheme vs. Our Scheme

since no attacker can derive the secrets preinstalled in the uncompromised sensors. Thus, our scheme is perfectly secure against the random node capture attack during the first phase of key establishment procedure. During the second phase of key establishment procedure, a unique key is assigned to each node for each of its neighboring zones. Thus, the attacker cannot derive more information from the captured information. Hence, for the second phase of our key establishment procedure, our proposed scheme is perfectly secure against random node capture attack. In summary, using our scheme, the attacker can not derive the secret information used among uncompromised nodes from the captured nodes. Overall, our scheme is resilient to random node capture attack. Figure 6 compares our scheme with the P-RKP location scheme proposed in [1] against selective attack.

8.4.3 Selective node capture attack

The selective node capture attack is described in Section 3.1. In this attack, the attacker can listen to the broadcast communications and selectively capture sensors to maximize the attack effects. Using selective node capture attack, the attacker only needs to attack certain area or a group of sensors. By doing this, with little efforts, a particular zone can be compromised.

The analysis of the fraction of compromised link among uncompromised nodes under selective attack on our scheme is the same that presented in previous section – the attacker cannot derive the keys used among uncompromised nodes from the captured nodes. But for P-RKP location scheme [1], if the attacker just concentrates on a particular area, he can compromise the system by capturing less number of nodes. As specified in [1], if the number of keys preinstalled for each sensor is $m = 50$ and the average number of neighbors for a node is $n' = 50$, the key pool size used by a particular zone is 1770. Note that the 1770 keys contain not only the keys for its own zone, but also keys in the neighboring zones (for details refer to [1]). Thus, by utilizing the (1) and attack techniques described in Section 3.1, we draw the Figure 6 to compare our scheme with P-RKP location scheme.

Figure 6 shows that under selective attack for P-RKP scheme, capturing 20 nodes will cause roughly 60% (1062 keys) of the total keys (1770) compromised. On the contrary, for our scheme, the fraction of compromised links between uncompromised nodes remains zero.

8.4.4 Node fabrication attack

Using node fabrication attack, the attacker can fabricate new nodes by manipulating the information from the captured nodes, such as the secret keys preinstalled in the captured sensors. This attack can cause severe security problems for P-RKP location scheme, since there is no connections between a sensor's *id* and the keys it possesses. For example, if an attacker only captures two nodes, he can fabricate $\binom{2m}{m}$ new nodes and deploy them back into the sensor network without being detected. Thus, the attacker can quickly outnumber the uncompromised sensors.

We use the secure group key schemes proposed by Blom [13] and further developed by Blundo et al [11], in which the key *id* is used to identify the row of the secret matrix A ($A = (D \cdot G)^T$)) distributed to the sensor (the key *id* can also serve as user *id*). Since we restrict the number of rows of the key matrix A distributed to sensors to λ, the attacker cannot derive the rows of key matrix A other than the ones he has already captured. Consequently, the attacker cannot fabricate new nodes using the information from the captured nodes.

8.4.5 Other security considerations

In other security attacks, such as node replication attack, cloned nodes can be deployed in the system. Our scheme's behavior is the same as that of existing proposals under this attack. Further, we notice that the attacker can compromise the pairwise keys by capturing sensors and then use the captured sensors to help the uncompromised sensors to setup the pairwise keys. This attack can be mitigated by increasing the probability that two sensors share a key p_1 and use multiple key paths to set up pairwise keys. The analysis of these countermeasures will be given in our following papers.

8.5 Communication Overhead Analysis

We derive the mathematical expressions for the probability that a sensor can set up key paths with all its neighbors within h hops in [14]. Due to page limits, we simply put the mathematical expressions, simulation results, and comparison studies in [14]. This mathematical model is used for our communication overhead analysis. 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 *key graph* connectivity probabilities: As shown in Figure 4, most pairwise keys can be set up within 3 hops. If a sensor cannot set up pairwise keys with its neighbors within 3 hops, as presented in Section 7.1, the sensor sends requests (r requests) to its neighbors, if q neighbors reply to the requests with the keys $k_1, \ldots, k_q$, where $q \leq r$, the pairwise key is $k = k_1 \oplus \ldots \oplus k_q$. We show that for $\omega = 7$ and $\tau = 2$, and if a sensor has 15 neighbors, then on

# of neighbors	# of hops	*key graph* 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

average, only 0.044 neighbors cannot set up pairwise keys within 4 hops. This is almost negligible.

During the second phase of key establishment, between two adjacent zones, as Figure 5(b) shows, a sensor can set up q paths to the neighboring zones with the probability of 0.9. Each path is a 2-hop path. The nodes who cannot find a path to the neighboring zone can use the method described in Section 7.2, which is the same as that in used in the first phase. In this scenario, a sensor sends requests (r requests) to its neighbors, if q neighbors reply to the requests with the keys $k_1, \ldots, k_q$, the pairwise key is $k = k_1 \oplus \ldots \oplus k_q$. Since the sensors within the curve lines (the area of the zone $\mathcal{Z}(i, j)$ is split by the curve that close to the neighboring zone) shown in Figure 5(b) has already set up the pairwise keys with all the neighbors in the neighboring zones, we only consider the nodes who can help the sensor to set up path with 2 hops. We note that the farther the sensor from the zone boundary, the smaller the probability that a sensor can find a path to the neighboring zone; at the same time, the farther the sensor from the boundary, the smaller the number of neighbors within the neighboring zone.

8.6 Computation Overhead Analysis

The computation overhead is mainly from the secure group key scheme introduced by SK-RKP scheme. In our schemes, we reduced the computation overhead significantly as compared to the SK-RKP scheme proposed in [8, 9] that does 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 [8, 9]). Note that this requirement is to fulfil the connectivity of whole sensor network. 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 sensor within a zone to $n_z = 100$, for $\omega = 7$, $\tau = 2$, the $\lambda = \lceil n_z \tau / \omega \rceil = 29$. Thus, the number of required modular multiplication operations to derive a pairwise key is only 58.

9. CONCLUSION

We described SK-RKP schemes [8, 9] used in our proposed grid-group scheme. Using this scheme, we significantly decrease the requirement for number of keys to be installed in each sensor (approximately 3 times less than the schemes without using deployment information). Moreover, our scheme is resilient to selective node capture attack and node fabrication attack which have not been completely addressed and analyzed in the current literatures. Our comparison studies with the recent P-RKP location scheme [1] show that our scheme exhibits better performance from both storage and security perspectives.

We notice that the attacker can compromise the pairwise keys by capturing sensors and then use the captured sensors to help (act as a proxy) the uncompromised sensors to setup the pairwise keys. In this way, the attacker can capture the pairwise keys used between uncompromised sensors. In our future work, we will study the security of various proposals under such attack and corresponding countermeasures.

ACKNOWLEDGEMENT

The authors would like to thank the anonymous reviewers for their valuable comments.

10. REFERENCES

[1] W. Du, J. Deng, Y. S. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *IEEE INFOCOM*, March 2004.

[2] D. Liu and P. Ning, "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)*, 2003, pp. 72 – 82.

[3] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: Mobile networking for "smart dust"," in *International Conference on Mobile Computing and Networking (MOBICOM)*, 1999, pp. 271–278.

[4] "Wireless integrated network sensors," University of California.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, pp. 102 – 114, August 2002.

[6] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of 9th ACM Conference on Computer and Communication Security (CCS-02)*, November 2002, pp. 41–47.

[7] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of 2003 Symposium on Security and Privacy*. Los Alamitos, CA: IEEE Computer Society, May 11–14 2003, pp. 197–215.

[8] D. Liu and P. Ning, "Establishing pairwise keys in distributed sensor networks," in *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*, October 2003, pp. 52–61.

[9] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," in *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*, October 2003, pp. 42–51.

[10] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "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'03)*, November 2003.

[11] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," *Information and Computation*, vol. 146, no. 1, pp. 1–23, 1998.

[12] R. D. Pietro, L. V. Mancini, and A. Mei, "Efficient and resilient key discovery based on pseudo-random key pre-deployment," in *18th International Parallel and Distributed Processing Symposium (IPDPS'04)*, April 2004.

[13] R. Blom, "An optimal class of symmetric key generation systems," in *EUROCRYPT'84*, ser. Lecture Notes in Computer Science, vol. 209. Paris, France: Springer-Verlag, 1985, pp. 335–338.

[14] D. Huang, M. Mehta, D. Medhi, and L. Harn, "Modeling pairwise key establishment for random key predistribution in large-scale sensor networks," University of Missouri – Kansas City, Tech. Rep., July 2004. [Online]. Available: http://conrel.sice.umkc.edu/dhuang/modeling.pdf

APPENDIX

A. SENSOR COVERAGE – WITHIN THE SAME ZONE

The covering area of sensor $[(i,j),b]$ in its zone $\mathcal{Z}(i,j)$ is shown in Figure 7. We can further divide the zone into 4 areas. The sensor coverage in these 4 areas are horizontally and vertically mapping to each other. Within a small area, there are two scenarios that shown in the Figure 7.

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 θ_1 plus two triangles (the shaded areas). The θ_1 is given as follows:

$$\theta_1 = \frac{3}{2}\pi - \sin^{-1}\frac{B_1 - x_1}{R} - \sin^{-1}\frac{A_1 - y_1}{R}$$

where,

$$A_1 - y_1 = \sqrt{R^2 - x_1^2}, \qquad B_1 - x_1 = \sqrt{R^2 - y_1^2}$$

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

$$\begin{aligned}\mathcal{C}_b^1(i,j)|_{(x_1,y_1)} &= x_1y_1 + \frac{1}{2}\left[x_1(A_1 - y_1)\right. \\ &\quad \left. + y_1(B_1 - x_1)\right] + \frac{\theta_1}{2}R^2\end{aligned}$$

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 θ_2 and θ_3. The θ_3 is given as follows:

$$\theta_3 = \frac{3}{2}\pi - \sin^{-1}\frac{B_2 - x_2}{R} - \sin^{-1}\frac{A_2 - y_2}{R}$$

The θ_2 is given as follows:

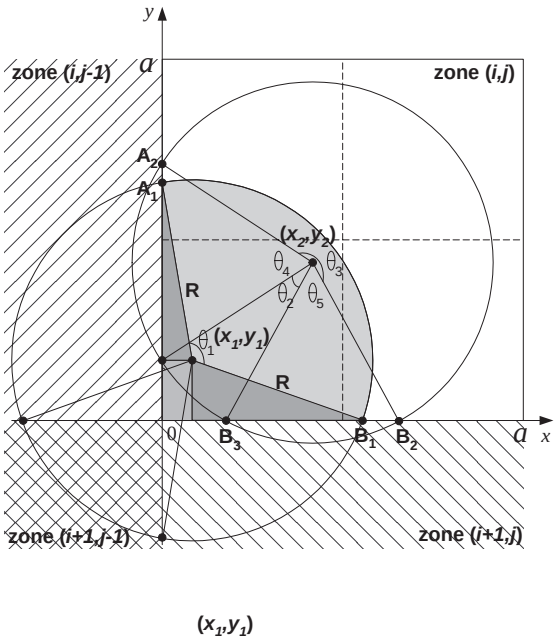
$$\theta_2 = \frac{\pi}{2} - \sin^{-1}\frac{x_2 - B_3}{R} - \sin^{-1}\frac{y_2 - A_3}{R}$$

The θ_4 is given as follows:

$$\theta_4 = 2\sin^{-1}\frac{A_2 - y_2}{R}$$

The θ_5 is given as follows:

$$\theta_5 = 2\sin^{-1}\frac{B_2 - x_2}{R}$$



(x_1, y_1)

Figure 7: Occupied zone

We note that,

$$y_2 - A_3 = A_2 - y_2 = \sqrt{R^2 - x_2^2}$$
$$x_2 - B_3 = B_2 - x_2 = \sqrt{R^2 - y_2^2}$$

The coverage (the shade area) of sensor $[(i,j),b]$ in zone (i,j) is represented as $\mathcal{C}_b^2(i,j)$ and it is computed as follows:

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

To summarize, given a position (x,y) for sensor $[(i,j),b]$, the area coverage is given as follows:

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

In Figure 4 shows the average number of neighbors of a sensor that is located in different position of a given zone.

B. SENSOR COVERAGE – IN DIFFERENT ZONE

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

Shown in Figure 8 (a), we first compute the area of the triangle $\triangle A_3 A_1 b$ as follows:

$$\begin{aligned} |A_1A_2| &= \sqrt{R^2 - x^2} \\ |A_3b| &= \frac{x}{\cos(\angle A_3bA_2)} = \frac{xR}{\sqrt{R^2-y^2}} \\ |A_2A_3| &= \frac{xy}{\sqrt{R^2-y^2}} \\ |A_2b| &= \sqrt{|A_3b|^2 - |A_2A_3|^2} \\ \triangle A_3A_1b &= \frac{1}{2}(|A_1A_2| + |A_2A_3|)|A_2b| \end{aligned}$$

The area of triangle $\triangle A_3 o B_1$ is given as follows:

$$\begin{aligned} |B_1A_3| &= R - |A_3b| \\ |A_3o| &= y - |A_2A_3| \\ |B_1o| &= \sqrt{|B_1A_3|^2 - |A_3o|^2} \\ \triangle B_1A_3o &= \frac{1}{2}|B_1o||A_3o| \end{aligned}$$

The area of sector $\widehat{B_1bA_1}$ is given as follows:

$$\begin{aligned} \angle A_2bA_3 &= \sin^{-1}\left(\frac{y}{R}\right) \\ \angle A_2A_1b &= \sin^{-1}\left(\frac{x}{R}\right) \\ \angle A_1bA_2 &= \frac{\pi}{2} - \angle A_2A_1b \\ \angle A_1bB_1 &= \angle A_1bA_2 + \angle A_2bA_3 \\ \widehat{B_1bA_1} &= \frac{\angle A_1bB_1}{2\pi}\pi R^2 = \frac{\angle A_1bB_1}{2}R^2 \end{aligned}$$

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

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

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

Shown in Figure 8 (b), we first compute the area of sector $\widehat{B_4bA_4}$:

$$\widehat{B_4bA_4} = \frac{\pi R^2}{4}$$

The area of sector $\widehat{B_1bA_4}$ is given as follows:

$$\begin{aligned} \angle B_1bA_4 &= \sin^{-1}\left(\frac{y}{R}\right) \\ \widehat{B_1bA_4} &= \frac{\angle B_1bA_4}{2}R^2 \end{aligned}$$

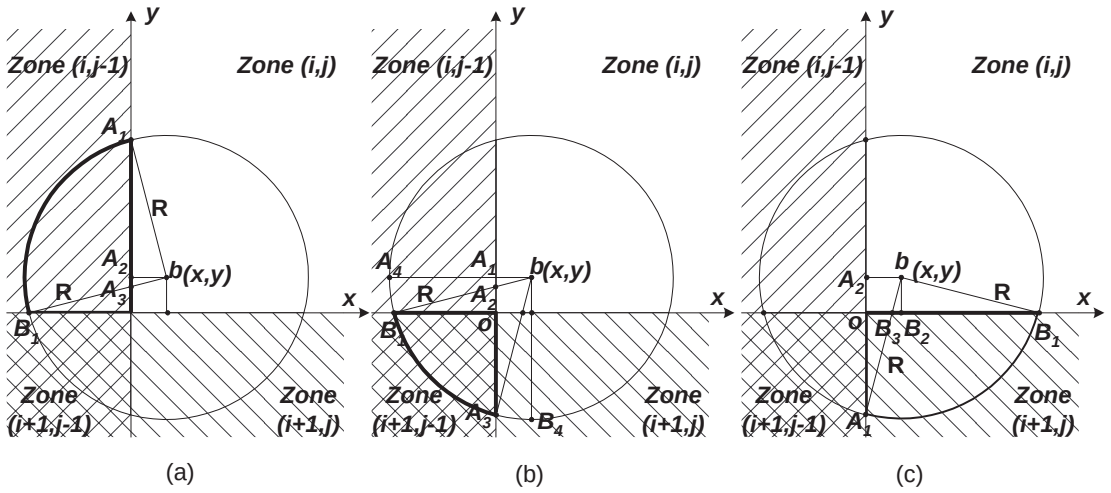


Figure 8: Zone Coverage Proof

The area of triangle $\triangle bB_1B_3$ is given as follows:

$$\begin{aligned} |B_1B_3| &= \sqrt{R^2-y^2} \\ \triangle bB_1B_3 &= \frac{1}{2}|B_1B_3|y \end{aligned}$$

The area of sector $\widehat{B_4bA_3}$ is given as follows:

$$\begin{aligned} \angle B_4bA_3 &= \sin^{-1}\left(\frac{x}{R}\right) \\ \widehat{B_4bA_3} &= \frac{\angle B_4bA_3}{2}R^2 \end{aligned}$$

The area of triangle $\triangle bA_1A_3$ is given as follows:

$$\begin{aligned} |A_1A_3| &= \sqrt{R^2-x^2} \\ \triangle bA_1A_3 &= \frac{1}{2}|A_1A_3|x \end{aligned}$$

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

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

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

Shown in Figure 8 (c), we first compute the area of triangle $\triangle A_3A_1b$ as follows:

$$\begin{aligned} |B_1B_2| &= \sqrt{R^2-y^2} \\ |B_3b| &= \frac{y}{\cos(\angle B_3bB_2)} = \frac{yR}{\sqrt{R^2-x^2}} \\ |B_2B_3| &= \frac{xy}{\sqrt{R^2-x^2}} \\ |B_2b| &= \sqrt{|B_3b|^2 - |B_2B_3|^2} \\ \triangle B_3B_1b &= \frac{1}{2}(|B_1B_2| + |B_2B_3|)|B_2b| \end{aligned}$$

The area of triangle $\triangle B_3oA_1$ is given as follows:

$$\begin{aligned} |A_1B_3| &= R - |B_3b| \\ |B_3o| &= x - |B_2B_3| \\ |A_1o| &= \sqrt{|A_1B_3|^2 - |B_3o|^2} \\ \triangle A_1B_3o &= \frac{1}{2}|A_1o||B_3o| \end{aligned}$$

The area of sector $\widehat{A_1bB_1}$ is given as follows:

$$\begin{aligned} \angle B_2bB_3 &= \sin^{-1}\left(\frac{x}{R}\right) \\ \angle B_2B_1b &= \sin^{-1}\left(\frac{y}{R}\right) \\ \angle B_1bB_2 &= \frac{\pi}{2} - \angle B_2B_1b \\ \angle B_1bA_1 &= \angle B_1bB_2 + \angle B_2bB_3 \\ \widehat{A_1bB_1} &= \frac{\angle B_1bA_1}{2\pi}\pi R^2 = \frac{\angle B_1bA_1}{2}R^2 \end{aligned}$$

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

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