An application-driven attack probability-based deterministic pairwise key pre-distribution scheme for non-uniformly deployed sensor networks

Chia-Mu Yu
Department of Electrical Engineering,
National Taiwan University,
Taipei 106, Taiwan
and
Institute of Information Science,
Academia Sinica,
Taipei 115, Taiwan
Email: r91045@csie.ntu.edu.tw

Cheng-Chung Li
Department of Computer Science and Information Engineering,
National Taiwan University,
Taipei 106, Taiwan
Email: r92087@csie.ntu.edu.tw

Chun-Shien Lu*
Institute of Information Science,
Academia Sinica,
Taipei 115, Taiwan
Email: lcs@iis.sinica.edu.tw
*Corresponding author

Sy-Yen Kuo
Department of Electrical Engineering,
National Taiwan University,
Taipei 106, Taiwan
Email: sykuo@cc.ee.ntu.edu.tw

Abstract: Secure communication is a challenging problem in wireless sensor networks. Key pre-distribution is commonly used to establish key sharing among sensor nodes. In this paper, we investigate an attack probability-based deterministic key pre-distribution method for non-uniform sensor deployment aiming mainly at preserving the integrity of the network to ensure the functionality of sensor networks. We first design a model for group-based deployment to estimate the attack probability of each deployment group. The proposed model further considers the motion of the adversaries and the location of the data sink to meet the practical situation. Based on this attack model, a deterministic key pre-distribution mechanism for group-based deployment is proposed. To evaluate the proposed method, the issues of memory overhead, communication overhead, and the fraction of total communication compromised after adversaries launch node capture attacks analytically and empirically. Simulation results show that our approach is superior than some known methods.

Keywords: attack probability; key pre-distribution; security; sensor network.

1 Introduction

1.1 Background

A Wireless Sensor Network (WSN) is composed of a large number of resource-constrained sensor nodes (Akyildiz et al., 2002). Usually, a WSN is deployed by an authority. The confidentiality in WSNs is usually achieved by symmetric cryptography, instead of asymmetric cryptography, because the latter consumes more energy and time in computation. Thus, it is essential to construct an efficient key distribution method for WSNs. Eschenauer and Gligor (2002) first proposed a probabilistic key pre-distribution scheme to achieve key sharing in WSNs. In their scheme, a key pool is first prepared for key distribution. Then, $S$ distinct keys, which are randomly chosen from $S$ and assigned to each sensor node, constitute a key ring of a sensor node. If two sensor nodes have at least one common key in their key rings, then they can communicate with each other using this common secret key.

The idea behind probabilistic key pre-distribution exploits the fact that a random graph achieves connectedness with high probability if each node’s degree is above a threshold. In this model, the sensor nodes and key sharing, respectively, correspond to the nodes and edges of a graph, called key graph. Note that the shortest path connecting two nodes in the key graph is called the key path. In addition, if a constraint is further imposed on a pair of sensor nodes that have key shared between them, i.e., these two sensor nodes are within the communication range of each other, then the key graph is called Kryptograph. Each edge between two nodes on the Kryptograph is called a secure link, which means that these two sensor nodes not only are within the communication range of each other but also share at least a common key. Each path between two nodes on the Kryptograph is called secure path, on which each edge corresponds to a secure link.

In contrast with probabilistic key pre-distribution, in deterministic key pre-distribution, the key graph is constructed in advance, and the pairwise keys are assigned to sensor nodes according to the edges in the key graph. Deterministic key pre-distribution is advantageous in reducing communication overhead because it averagely provides shorter key path by using smaller key rings (Çamtepe and Yener, 2007). A recent survey of different key pre-distribution schemes can be found in Xiao et al. (2007). On the other hand, the key distribution schemes in the literature usually aim at achieving key sharing without considering the underlying applications of WSNs, which mostly are object monitoring and environment surveillance. In such applications, two observations can be obtained. First, certain knowledge of the sensing region can be obtained in advance. For example, some locations, which are likely to be intruded, can be known in advance from, say, official statistics, in the urban sensing applications. However, such observation is rarely utilised in the previous key distribution schemes. Second, the integrity of the networks is extremely important in environment surveillance because the detection of objects is unknown to the authority as long as there is no path connecting the nodes, which detect objects, and the data sink.
1.2 Our contribution

In this paper, an attack probability-based deterministic key pre-distribution method for non-uniform sensor deployment is proposed. The major contributions include:

1. the design of a reasonable model for group-based deployment to estimate the attack probability of each deployment group
2. the proposed model is the first to take the location of the data sink into consideration to meet the practical situation
3. based on this attack model, we propose a deterministic key pre-distribution mechanism for group-based deployment, which leads to more precise indication of the number of communication influenced by a node capture attack launched by the adversary.

In addition to the use of deployment knowledge and further study of attack probability, we also investigate the following issues: (a) comparison of memory overhead, local connectivity, and global connectivity; (b) the relationship between the size of component consisting of data sink and number of captured sensors; (c) the relationship between the fraction of compromised communication and number of captured sensors; and (d) the relationship between the expected communication overhead and number of captured sensors.

The remainder of this paper is organised as follows. In Section 2, the system model is described. In Section 3, the proposed method, including the attack probability model and attack probability-based deterministic key pre-distribution, is described. Our analytical and simulation results are, respectively, elaborated in Sections 4 and 5. Some related works on key distribution methods and security protocols are described in Section 6. Finally, conclusions are given in Section 7.

2 System model

Let the total number of sensor nodes to be deployed be N. The sensor nodes are divided into k subgroups, G_i, for 1 ≤ i ≤ κ, whose number of sensor nodes are n_i, i.e. N = \sum_{i=1}^{κ} n_i. Each subgroup is called a deployment group.

For each deployment group, we choose a deployment point, which is defined as the location where sensor nodes are scattered. Different from the deployment point, the resident point is defined as the location where a sensor node is finally located. In this paper, the probability density function for resident points is assumed to follow a two-dimensional Gaussian distribution, i.e.

\[ \text{Gau}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_c)^2+(y-y_c)^2}{2\sigma^2}}, \tag{1} \]

where \( \sigma \) is the standard deviation and (\( x_c, y_c \)) is the deployment point of G_i. However, it should be noted that our proposed method can also be applied to other probability density functions such as uniform distribution.

In this paper, the sensing region is assumed to be divided into grids for simplifying analyses. In other words, the deployment points are arranged in a two-dimensional grid, as shown in Figure 1, where the deployment point of group G_{ij} is located at the i-th row and j-th column.

From the experience of sensor network design, the following assumptions can be made with respect to the communication pattern obtained for the group-based deployment (Du et al., 2004):

1. if two sensor nodes are within the same deployment group, then communication between them frequently happens
2. communication between two sensor nodes that is, respectively, located in two horizontally/vertically adjacent deployment groups are less frequent than communication within the same deployment group
3. communication happens rarely if two sensor nodes are positioned in diagonally adjacent deployment groups
4. compared with communication between adjacent deployment groups, there is almost no communication between two deployment groups that are non-adjacent.

3 Proposed method

Based on group-based deployment model, Chan et al. (2006) further expressed that, for group-based deployment, each deployment group is associated with an attack probability, which can be exploited to improve the resistance to node capture attack. However, the attack probability for each deployment group in their paper was predefined to be a constant without the possibility of depending on other groups. In other words, they assume that the exact attack probabilities of sensor nodes in different groups are known a priori. Moreover, according to Chan et al. (2006) because each group is associated with a different attack probability, the key ring size of each group should be different in order to enhance
security and efficient memory usage of sensor nodes. For example, if a group is associated with a higher attack probability, then this group naturally needs a larger key ring size to increase the probability of key sharing between a pair of sensor nodes. However, a larger key ring size implies lower resilience so that the key ring size must be chosen carefully to achieve the trade-off. Hence, one of our goals in this paper is to present a reasonable model to calculate the attack probability of each group, rather than use predefined meaningless attack probabilities.

The proposed method not only exploits the frequency of communication between two deployment groups to enhance key usage and reduce communication overhead, but also takes advantage of different attack probabilities of deployment groups, which are derived from the attack probability model we develop, to improve the resistance of sensor nodes to node capture attack. The notations used throughout this paper are summarised as Table 1.

Table 1 Nomenclature

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>The total number of sensor nodes</td>
</tr>
<tr>
<td>n</td>
<td>The number of sensor nodes in a deployment group</td>
</tr>
<tr>
<td>Gi,j</td>
<td>The deployment group with deployment point (x, y)</td>
</tr>
<tr>
<td>Dij</td>
<td>The deployment point of group Gi,j</td>
</tr>
<tr>
<td>bij,bi,j</td>
<td>Base coefficients</td>
</tr>
<tr>
<td>αi,αi,j</td>
<td>Descendent factors</td>
</tr>
<tr>
<td>gi,gi,j</td>
<td>Influence factors</td>
</tr>
<tr>
<td>DS</td>
<td>The set of data sinks</td>
</tr>
<tr>
<td>GDS</td>
<td>The set of deployment groups consisting of the data sink</td>
</tr>
<tr>
<td>ωi,j</td>
<td>Attack coefficient</td>
</tr>
<tr>
<td>pi,j</td>
<td>Attack probability</td>
</tr>
<tr>
<td>Di</td>
<td>Intra-group node degree determination function</td>
</tr>
<tr>
<td>di,j</td>
<td>Intra-group node degree in Gi,j, i.e. (d_{ij} = D_i(p_{i,j}))</td>
</tr>
<tr>
<td>DC</td>
<td>Inter-group node degree determination function</td>
</tr>
<tr>
<td>(\hat{d}_{(i,j)(i',j')})</td>
<td>Inter-group node degree between Gi,j and Gi',j', i.e. (\hat{d}<em>{(i,j)(i',j')} = D_C(p</em>{i,j}, p_{i',j'}))</td>
</tr>
<tr>
<td>plocal</td>
<td>Local connectivity</td>
</tr>
<tr>
<td>pglobal</td>
<td>Global connectivity</td>
</tr>
<tr>
<td>C_{rs}</td>
<td>A circular area centred at a sensor node s with radius r</td>
</tr>
<tr>
<td>C_{rθ}</td>
<td>An area which is the union of circles centred at the sensor nodes from (\Theta) with radius r, where (\Theta) is a set of sensor nodes</td>
</tr>
<tr>
<td>Er,s</td>
<td>The sensor nodes within (C_{rs})</td>
</tr>
<tr>
<td>Er,θ</td>
<td>The sensor nodes within (C_{rθ})</td>
</tr>
</tbody>
</table>

3.1 Basic idea

In many applications, we can know that some specific locations are liable to be intruded by the adversary. For example, the bank is more likely to be intruded than the park even when they are neighbours. Based on this observation, it is known that deploying sensor nodes uniformly over the sensing region is undesirable. On the contrary, the sensor nodes should be densely deployed over the specific locations that are liable to be intruded. The remaining problem is how to determine the exact attack probability for each location. Note that due to the spatial-temporal characteristic of the physical object, which means the adversary in this paper, the initial relative attack probability of a location will be diffused to its neighbouring locations since the adversary will tend to attack neighbouring locations instead of a single location only. Based on this observation, the exact attack probability for each location should be calculated by considering cross-influence of the relative attack probabilities of neighbouring locations. After having the attack probabilities assigned to different locations, proper number of secret keys can then be assigned to the sensor nodes of different locations.

3.2 Attack probability estimation

In this section, we develop a reasonable model to formally derive the attack probability for each group. Consider that sensor nodes are deployed in terms of group-based deployment, as shown in Figure 1. The attack probability of a deployment group refers to the probability that a deployment group is attacked by adversaries. Generally, the attack probability cannot obviously be obtained and is dependent on those of other deployment groups. In this study, we also consider that a sensor node can play a role of a data sink. Basically, the attack probability of a deployment group depends on three factors, which are (a) the base coefficient; (b) the influence of neighbouring deployment groups; and (c) the location of the data sink.

In WSNs, each deployment group Gi,j is associated with a base coefficient bij representing the degree of threat from adversaries. The base coefficients depend on applications and can be estimated via prior knowledge. For example, if the communication overhead of Gi,j is heavier than those of other groups, then Gi,j tends to be attacked with higher probability. Under this circumstance, Gi,j must be associated with a higher attack coefficient.

In addition, when adversaries launch an attack, not only the sensor nodes in a target deployment group but also the sensor nodes of those groups neighbouring the target one will be attacked. This is based on the concept of the spatio-temporal continuity of physical objects (Galton, 2001), which means that every physical object’s movement, such as the movement of humans and cars, is continuous. More specifically, when the adversary leaves the currently monitored region, he/she will move to the new region that is
neighbouring to the currently monitored region. Therefore, a descendent factor \( \alpha \) is used to indicate how many tiers of neighbouring groups will be also affected. For deployment group \( G_{i,j} \), these \( \alpha \) tiers of neighbouring groups are associated with the attack influence factors \( g_i b_{i,j}, g_2 b_{i,j}, \ldots, g_n b_{i,j} \), respectively, where \( 0 \leq g_1, g_2, \ldots, g_n \leq 1 \).

For the third factor, since the data sink is most likely to be attacked, the attack probability originating from the data sink has to be taken into consideration. This is because the data sink is the only gateway of forwarding collected data from sensor nodes to the authority. The easiest way for adversaries to break a WSN is to disconnect the links from sensor nodes to the data sink, leading to the highest pay-off. Therefore, the attack probability of a deployment group containing the data sink, \( \mathbf{DS} \), is defined as

\[
\omega_{i,j} = b_{i,j} + \sum_{\rho=1}^{\alpha} \sum_{i' \neq i} (g_{i'} b_{i',j} + g_{i'} b_{i',j}),
\]

where \( \gamma(G_{i,j}, \rho) = \{ (b_{i,j}, \tilde{b}_{i,j}) : |i' - i| = \rho \text{ or } |j' - j| = \rho \} \) is a set of pairs of the base coefficient \( b_{i',j} \) for \( G_{i,j} \) and base coefficient \( \tilde{b}_{i,j} \) for data sink satisfying that \( G_{i,j} \) and \( G_{i',j} \) are at a distance from \( \rho \) deployment groups.

From equation (2), it can be seen that the deployment containing a data sink possesses higher attack probabilities since \( \tilde{b}_{i,j} \neq 0 \).

2 The normalised attack probability, \( p_{i,j} \), with respect to the deployment group \( G_{i,j} \), is defined as

\[
p_{i,j} = \frac{\omega_{i,j}}{\sum_{i',j'} \omega_{i',j'}}.
\]

Figure 1 illustrates a simple example, where \( b_{3,4} = 10, b_{5,5} = 1, \) and \( b_{6,5} = 5. \) If \( \alpha = 2 \) and the influence factors are \( \{g_1, g_2\} = \{0.5, 0.1\} \), for each deployment group, the effect of the attack will be diffused to its neighbouring deployment groups whose locations are within two tiers around it. For example, in Figure 1, \( G_{3,4} \) contributes an attack influence factor \( b_{3,4} \times g_1 = 5 \) to \( G_{5,5} \) because \( G_{2,5} \) is the first-tier neighbour of \( G_{3,4} \). Similarly, \( G_{5,5} \) contributes \( 0.5 \) to \( G_{5,6} \), and \( G_{6,5} \) contributes \( 0.5 \) to \( G_{5,6} \). Since the distance between \( G_{5,5} \) and \( G_{5,6} \) is beyond the descendent factor \( \alpha \), they do not influence each other. Moreover, from the viewpoint of data sink, the only deployment group containing a data sink is \( G_{5,6} \). Thus, \( G_{5,6} \) also diffuses the attack influence factors to its neighbouring deployment groups. For example, if \( \tilde{b}_{5,6} = 8, \tilde{a} = 2, \) and \( \{\tilde{g}_1, \tilde{g}_2\} = \{0.5, 0.1\} \), then \( G_{5,6} \) contributes \( \tilde{b}_{5,6} \times \tilde{g}_2 \) to \( G_{5,4} \) because \( G_{5,4} \) is the second-tier neighbour of \( G_{5,6} \) containing a data sink, the descendent factor, \( \tilde{a} \), for data sink and attack influence factors, \( \{\tilde{g}_1 b_{1,j}, \ldots, \tilde{g}_n b_{n,j}\} \), where \( 0 \leq \tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_n \leq 1 \), for data sink are chosen.

The overall attack probability of a deployment group can be formulated as follows:

1 The so-called attack coefficient \( \omega_{n,j} \), associated with \( G_{i,j} \), is a value gained by considering all of the factors discussed above. Thus, \( \omega_{n,j} \) can be determined as

\[
\omega_{i,j} = b_{i,j} + \sum_{\rho=1}^{\alpha} \sum_{i' \neq i} (g_{i'} b_{i',j} + g_{i'} b_{i',j}),
\]

where

\[
\gamma(G_{i,j}, \rho) = \{ (b_{i,j}, \tilde{b}_{i,j}) : |i' - i| = \rho \text{ or } |j' - j| = \rho \}
\]

is a set of pairs of the base coefficient \( b_{i',j} \) for \( G_{i,j} \) and base coefficient \( \tilde{b}_{i,j} \) for data sink satisfying that \( G_{i,j} \) and \( G_{i',j} \) are at a distance from \( \rho \) deployment groups.

From equation (2), it can be seen that the deployment containing a data sink possesses higher attack probabilities since \( \tilde{b}_{i,j} \neq 0 \).

2 The normalised attack probability, \( p_{i,j} \), with respect to the deployment group \( G_{i,j} \), is defined as

\[
p_{i,j} = \frac{\omega_{i,j}}{\sum_{i',j'} \omega_{i',j'}}.
\]

Let \( DS \) be a set of sensor nodes, which are data sinks. Let \( DS \) be a set of deployment groups containing at least one data sink. Given \( G_{i,j} \in DS \), the base coefficient, \( \tilde{b}_{i,j} \), for data sink is assigned, which should be different from the base coefficient \( b_{i,j} \), \( \tilde{b}_{i,j} = 0 \) if \( G_{i,j} \notin DS \) and \( \tilde{b}_{i,j} \neq 0 \) if \( G_{i,j} \in DS \). Furthermore, for the deployment group \( G_{i,j} \), the set of sensor nodes \( S_{i,j} \) is defined as

\[
S_{i,j} = \{ s_1, s_2, s_3 \}.
\]

An illustration is shown in Figure 2, where a snapshot of a network with single data sink is considered. To break down the networks, in addition to directly making the data sink crash, the easiest way is to capture the set of sensor nodes \( \{ s_1, s_2, s_3 \} \). If capturing these three sensor nodes is not feasible for certain reasons, then the secondary choice for the adversary is to capture the set of sensor nodes \( \{ s_1, s_2, s_3 \} \). Therefore, a simple conclusion is that the attack probability of the sensor nodes closer to the data sink should be higher than the attack probability of the sensor nodes which are distant from the data sink.

Figure 2 Increment of attack probability for those sensor nodes that are near the data sink (see online version for colours)
3.3 Attack probability-based deterministic key pre-distribution

The proposed attack probability-based deterministic key pre-distribution mechanism is described in this section. The proposed method not only exploits the frequency of communication between two deployment groups to enhance key usage and reduce communication overhead, but also takes advantage of different attack probabilities of deployment groups to improve the resistance of sensor nodes to node capture attack.

Our method is composed of four phases: (a) area partitioning phase; (b) key pre-distribution phase; (c) shared key discovery phase; and (d) path key establishment phase. During the area partitioning phase, the deployment point and attack probability corresponding to each deployment group are determined. In the key pre-distribution phase, keys are stored in sensor nodes according to both the prior knowledge (to be discussed later) obtained from the area partitioning phase and the estimated attack probability for each deployment group. Both the area partitioning phase and the key pre-distribution phase are performed in an off-line manner, i.e. they are performed before sensor nodes are deployed.

On the other hand, both the shared key discovery phase and the path key establishment phase are used for the pair of sensor nodes which would like to communicate to find a common key. When two sensor nodes want to communicate and have a common key within their own key rings, the shared key discovery phase is performed. Otherwise, if two sensor nodes willing to communicate with each other do not have a common key within their own key rings, then the path key establishment phase is performed to construct a common key indirectly. Both the shared key discovery phase and the path key establishment phase are performed in an on-line manner, i.e. they are performed after sensor nodes have been deployed.

In the following sub-sections, the four phases will be described in detail.

3.3.1 Area partitioning phase

In this phase, two goals must be achieved. First, under the consideration of the total number of sensor nodes, the size of sensing region, and the variance of the locations of sensor nodes in a deployment group, the total number of deployment groups and the locations of their corresponding deployment points need to be calculated such that the resultant locations of sensor nodes form nearly a uniform distribution.\(^4\) The reason for achieving a nearly uniform distribution is so that the coverage of the sensing region can be similarly equal, avoiding a detection hole in the sensing region. After these parameters have been determined, the number of sensor nodes within each deployment group can also be determined. Second, the attack probability for each deployment group needs to be calculated, as described in Section 3.2. Now, we only explain how we achieve the first goal in detail as follows.

Given that the total number of sensor nodes is \(N\), after the region to be monitored is determined, the shape and size of this sensing region are determined accordingly. Without loss of generality, we assume that the shape of the sensing region is a square of size \(a \times a\), and the deployment points will be arranged in a two-dimensional grid. If group-based deployment model is considered, the variance \(\sigma^2\) of sensor nodes in each deployment group [shown in equation (1)] is also determined. Since approximately 95.4% of sensor nodes from one deployment group will be scattered within the circular area centred at the deployment point with radius \(2\sigma\), we suggest that the distance between the deployment points for two neighbouring deployment groups should be \(2\sigma\). Thus, by a simple calculation, it can be obtained that there should be \(\left\lfloor \frac{a-2\sigma}{2\sigma} \right\rfloor \times \left\lfloor \frac{a-2\sigma}{2\sigma} \right\rfloor\) deployment groups, and the corresponding locations \((x_i, y_j)\) of deployment points for the deployment group \(G_{i,j}\) can also be calculated as:

\[
(x_i, y_j) = (\sigma + (i-1) \cdot 2\sigma, \sigma + (j-1) \cdot 2\sigma).
\]

Finally, the number of sensor nodes for each deployment group can be calculated as:

\[
N = \left\lfloor \frac{a-2\sigma}{2\sigma} \right\rfloor \times \left\lfloor \frac{a-2\sigma}{2\sigma} \right\rfloor
\]

3.3.2 Key pre-distribution phase

The goal of this phase is to store keys into each sensor node according to the estimated attack probability of each deployment group. For each deployment group \(G_{i,j}\), we construct the key graph based on a random regular graph \(R_{d_{i,j}}\) using the algorithm presented by Steger and Wormald (1999), where \(n\) denotes the number of sensor nodes that is the same for all deployment groups and \(d_{i,j}\) denotes the degree of each sensor node in \(G_{i,j}\). The major difference distinguishing our method from the method of Chan et al. (2006) is that in our method \(d_{i,j}\) is primarily determined based on the attack probability of \(G_{i,j}\). In fact, the higher probability that a deployment group of being attacked implies that \(d_{i,j}\) should be set to be a higher value. That is, for each deployment group, the degree of a node in the key graph should be positively proportional to its attack probability. In this paper, we discuss the problem of attack probability-based deterministic key pre-distribution from two aspects, i.e. intra-group degree determination and inter-group degree determination.

An intuitive principle behind intra-group degree determination is that the attack probability is positively proportional to the degree of key sharing. As shown in equation (4), let \(p_{i,j}\) and \(p_{i,j'}\), respectively, denote the
normalised attack probability of $G_{i,j}$ and $G_{i,j}$. Let $D_i(p_{i,j})$ be the intra-group degree determination function used to determine the degree of a node within $G_{i,j}$, i.e. $d_{i,j} = D_i(p_{i,j})$. In this paper, $D_i(p_{i,j})$ is empirically designed to be a threshold function. Let $\{\sigma_1, \sigma_2, \ldots, \sigma_{\Omega+1}\}$ be a set of threshold values and let $\{\pi_1, \pi_2, \ldots, \pi_{\Omega+1}\}$ be a set of $(\Omega + 1)$ integer values used to represent the degree values. The values $\{\pi_1, \pi_2, \ldots, \pi_{\Omega+1}\}$ are given such that, after the assignment of keys based on the above setting, sensor nodes can resist attacks in the corresponding groups they belong to. In practice, these values are designated empirically in order to match the memory size of sensor nodes. When $D_i()$ takes input as $p_{i,j}$, it maps $p_{i,j}$ into one of $(\Omega + 1)$ values. Formally, $D_i()$ can be represented as:

$$d_{i,j} = D_i(p_{i,j}) = \begin{cases} 
\pi_1, & \text{if } p_{i,j} < \sigma_1 \\
\pi_2, & \text{if } \sigma_1 \leq p_{i,j} < \sigma_2 \\
\vdots & \vdots \\
\pi_{\Omega+1}, & \text{if } \sigma_{\Omega+1} \leq p_{i,j}.
\end{cases}$$

(7)

where $\pi_1 \leq \pi_2 \leq \ldots \leq \pi_{\Omega+1}$. Therefore, if $p_{i,j} \leq p_{r,f}$ holds then $d_{i,j} \leq d_{r,f}$ holds.

Based on the group-based deployment model, the level of key sharing between two deployment groups, $G_{i,j}$ and $G_{i,f}$, is determined as follows. Let $d_{i,j}^*(p_{i,j}, p_{r,f})$ denote the needed node degree to achieve connection between $G_{i,j}$ and $G_{i,f}$. The value of $d_{i,j}^*(p_{i,j}, p_{r,f})$ is determined by $D_i(p_{i,j})$, which is the inter-group degree determination function used to determine the degree between $G_{i,j}$ and $G_{i,f}$ based on the maximum degree between them as:

$$d_{i,j}^*(p_{i,j}, p_{r,f}) = \begin{cases} 
\phi_1 \cdot \max\{d_{i,j}, d_{r,f}\}, & \text{if } G_{i,j} \cap G_{i,f} \\
\phi_2 \cdot \max\{d_{i,j}, d_{r,f}\}, & \text{if } G_{i,j} \cap G_{i,f} \\
\phi_3 \cdot \max\{d_{i,j}, d_{r,f}\}, & \text{if } G_{i,j} \cap G_{i,f}
\end{cases}$$

(8)

where $\phi_1, \phi_2$ and $\phi_3$ are within $[0,1]$ and they satisfy with $\phi_1 \leq \phi_2 \leq \phi_3$. Note that $d_{i,j}^*(p_{i,j}, p_{r,f})$ is equivalent to $D_i(p_{i,j}, p_{r,f})$. In addition, in equation (8), $G_{i,j} \cap G_{i,f}$ means that $G_{i,j}$ and $G_{i,f}$ are vertically or horizontally adjacent, $G_{i,j} \cap G_{i,f}$ means that $G_{i,j}$ and $G_{i,f}$ are diagonally adjacent, and $G_{i,j} \cup G_{i,f}$ means that $G_{i,j}$ and $G_{i,f}$ are non-adjacent. Similar to the determination of the intra-group degree determination function $D_i(p_{i,j})$, the inter-group degree determination function $D_i(p_{i,j}, p_{r,f})$ is determined by a heuristic, and the performances from different heuristics used will be presented in Section 5.2. Obviously, inter-group degree satisfies a symmetric property, i.e. $\hat{d}_{i,j}^*(p_{i,j}, p_{r,f}) = \hat{d}_{i,j}^*(p_{i,j}, p_{r,f})$.

Therefore, we can derive $\hat{d}_{i,j}^*(p_{i,j}, p_{r,f}) \leq \hat{d}_{i,j}^*(p_{i,j}, p_{r,f})$ if and only if $|i - n_i| + |j - n_j| = |i - n_i| + |j - n_j|$.

The parameters used in equations (7) and (8) will be empirically determined via simulations and discussed in Section 5.2.

### 3.3.3 Shared key discovery phase

The goal of this phase is to make two sensor nodes become connected with an edge in the key graph for finding the common key. This can be easily achieved because the key graph is constructed deterministically and the edges of key graph are pre-assigned by authority. Thus, when the source node $s$ wants to communicate with the destination node $d$, the source node $s$ first checks whether it has a key shared with the destination node $d$. If such a shared key exists, $s$ and $d$ use it to secure their communication. Otherwise, the path key establishment phase described in the following must be performed in order to agree on a new key between them.

### 3.3.4 Path key establishment phase

In this phase, a path key will be found between the pair of sensor nodes that do not share a key within their respective key rings. More precisely, if two sensor nodes $s$ and $d$ willing to communicate with each other can not find a shared key within their respective key rings, they try to find a path connecting them, on which each pair of sensor nodes has a shared key. Once such a path is found, the two ends $s$ and $d$ on this path can negotiate a path key using this secure path. Note that two methods can be used to generate the path key. The first one is that a new key, $k_{s,d}$, is randomly generated by one of the two ends on the secure path and then sent to the other end. The path key serves as the communication key between $s$ and $d$. The second one is that one of two ends on the secure path, for example $s$, selects one of recently unused keys as the common key, $k_{s,d}$, between $s$ and $d$.

### 4 Analytical results

In this section, we analytically investigate the issues of memory overhead, communication overhead, and the fraction of total communication compromised after adversaries launch node capture attacks for the proposed method.

#### 4.1 Memory overhead

Based on equations (7) and (8) defined in the key predistribution phase, the memory overhead of each sensor node in $G_{i,j}$ is its total degrees of sensor nodes, i.e.

$$D_i(p_{i,j}) + \sum_{(r,s) \neq j} D_i(p_{r,j}, p_{r,f}) = d_{i,j} + \sum_{(r,s) \neq j} \hat{d}_{i,j}^*(p_{r,j}, p_{r,f}).$$

(9)
This implies that higher degrees of sensor nodes can result in lower communication overhead. In addition, the capability of maintaining network integrity after node capture attack has been launched can also be enhanced by increasing degrees of sensor nodes because this implies that the possibility of having more node-disjoint secure paths between two sensor nodes is increased. Therefore, from this formulation, the trade-off among the memory overhead, the capability of maintaining network integrity after node capture attack, and communication overhead can be leveraged according to the limited storage space. A comparison between memory overhead and connectivity of our method will be shown in Section 5.3.

4.2 Connectivity

Two kinds of connectivity, global connectivity and local connectivity, are considered here. Global connectivity is defined to measure the utilisation ratio of all sensor nodes in a WSN. When global connectivity is \( r \% \), it means that \( 1 - r \% \) of sensor nodes are isolated from the maximum component of Kryptograph, where the maximum component is a maximally connected sub-graph of a graph. On the other hand, local connectivity refers to the probability that any two neighbouring sensor nodes have a common key. In the following, local connectivity is calculated using a technique similar to (Du et al., 2004).

Given two sensor nodes, \( s_1 \in G_{ij} \) and \( s_2 \in G'_{ij} \), local connectivity, \( p_{local} \), can be formulated as:

\[
p_{local} = \frac{Pr(KS(s_1, s_2))}{Pr(RC(s_1, s_2))},
\]

where \( KS(s_1, s_2) \) denotes the event that \( s_1 \) and \( s_2 \) have a common key and \( RC(s_1, s_2) \) denotes the event that \( s_1 \) (or \( s_2 \)) is within the communication range of \( s_2 \) (or \( s_1 \)). In order to compute \( p_{local} \), we first need to calculate \( Pr(KS(s_1, s_2) \cap RC(s_1, s_2)) \) and \( Pr(RC(s_1, s_2)) \).

Assume that there is an infinitesimal rectangular area of size \( dx \ dy \) centred at \( Z = (x, y) \). Let \( dis_{ij}(s, Z) \) be the distance between the sensor node \( s \) and the deployment point of \( G_{ij} \), where \( s \) is located within \( Z \). Let \( f_1(dis_{ij}, s, Z) \) be the probability density function for deployment group \( G_{ij} \) with respect to the distance between \( Z \) and the deployment point of \( G_{ij} \), i.e.:

\[
f_1(dis_{ij}, Z \parallel s) = \frac{1}{2\pi \sigma^2} e^{-\frac{(dis_{ij}(s, Z))^2}{2\sigma^2}}.
\]

An illustration of equation (11) is shown in Figure 3.

Similarly, the probability that the sensor node \( s_2 \), belonging to \( G'_{i'j'} \) with deployment point \( (x', y') \), resides within \( Z \) is

\[
f_2(dis_{i'j'}, s_2, Z) \parallel s_2 \in G'_{i'j'} \approx \frac{1}{2\pi \sigma^2} e^{-\frac{(dis_{i'j'}(s_2, Z))^2}{2\sigma^2}} dx \ dy
\]

An illustration of equation (12) is shown in Figure 4. Note that the term ‘\( dx \ dy \)’ of equation (12) is merely a trick used to calculate the probability of \( Pr(RC(s_1, s_2)) \), as will later be shown in equation (15).
sensor node $s_1$ is from $G_{i,j}$, $s_1$ resides within the circle centred at $Z$ with radius $r$. Here, $Z$ is defined to be an infinitesimal area containing $s_2$ and is located within $G_{i,j}$. $f_s(Z, r | s_1 \in G_{i,j})$ can be represented as:

$$f_s(Z, r | s_1 \in G_{i,j}) = \int_{x=r}^{x=\infty} \int_{y=0}^{y=\infty} \frac{1}{2\pi r^2} e^{-\frac{(x-r)^2 + y^2}{2r^2}} dxdy, \quad (13)$$

where $(x_r, y_r)$ is the deployment point of $G_{i,j}$. An illustration of equation (13) is shown in Figure 5.

**Figure 5** Schematic expression of equation (13)

Based on the above derivations, the product of functions $f_2$ [equation (12)] and $f_3$ [equation (13)] denotes the probability of $RC(s_1, s_2)$ at the specified deployment groups and locations of deployment points. As a result, $Pr(RC(s_1, s_2))$ of equation (10) can be calculated through the combination of functions $f_2$ and $f_3$ by considering all possible deployment groups and locations as:

$$Pr(RC(s_1, s_2)) = \int_{x=0}^{x=\infty} \int_{y=0}^{y=\infty} \sum_{G_{i,j} \in \Psi} \sum_{s_1 \in G_{i,j}} Pr(s_1 \in G_{i,j}) \cdot Pr(s_2 \in G_{i,j}) \cdot f_2(dis_{i,j}, (x_r, y_r)) \cdot f_3(Z, r | s_1 \in G_{i,j}) dxdy, \quad (14)$$

where $\Psi$ denotes the set of all deployment groups and $Pr(s_1 \in G_{i,j})$ represents the probability that $s_1$ is from $G_{i,j}$. In addition, if the average probability of key sharing between sensor nodes, denoted as $\frac{\hat{d}_{i,j}(s_1)}{n}$, is considered, then $Pr(KS(s_1, s_2) \cap RC(s_1, s_2))$ of equation (10) can be derived as:

$$Pr(KS(s_1, s_2) \cap RC(s_1, s_2)) = \int_{x=0}^{x=\infty} \int_{y=0}^{y=\infty} \sum_{G_{i,j} \in \Psi} \sum_{s_1 \in G_{i,j}} Pr(s_1 \in G_{i,j}) Pr(s_2 \in G_{i,j}) \cdot f_2(dis_{i,j}, (x_r, y_r)) \cdot f_3(Z, r | s_1 \in G_{i,j}) dxdy, \quad (15)$$

The numerical result regarding global connectivity and local connectivity will be presented in Section 5.3.

### 4.3 Communication overhead

First, we study the expected communication overhead needed for any two distinct sensor nodes, $s_1$ and $s_2$, within the same deployment group $G_{i,j}$. In this case, we denote $X$ as a random variable, which represents the communication overhead between $s_1$ and $s_2$, and $E[X]$ as the expected communication overhead between any pair of sensor nodes in the same deployment group. $E[X]$ can be written as $E[X] = 1 \cdot p_1 + 2 \cdot p_2 + \cdots + \sum_{h=2}^{\infty} h \cdot p_h$, where $p_h$ is the probability that given a pair of sensor nodes $s_1$ and $s_2$ the number of hops required for sending messages from $s_1$ to $s_2$ is $h$. To calculate $p_h$, we also need the following information. Define $E_{r, h} = \{s \mid s \in C_{r, h}, s \not\in \Theta\}$ as a set of sensor nodes lying within the circles from $C_{r, h}$ but not coming from $\Theta$. Moreover, recursive definitions of $C_{r, h}$ and $E_{r, h}$ for all $h$’s are defined as:

$$E^{(1)}_{r, h} = E_{r, h}, E^{(2)}_{r, h} = E_{r, h, h} \cap \bigcup_{h=2}^{\infty} E^{(h)}_{r, h}, \quad (17)$$

and

$$C^{(1)}_{r, h} = C_{r, h}, C^{(2)}_{r, h} = C_{r, h, h} \cap \bigcup_{h=2}^{\infty} C^{(h)}_{r, h}, \quad (18)$$

respectively. Note that, if sensor node $s_1$ is located within $\Theta$ and sensor node $s_2$ is located within the area $C_{r, h}$, then they are exactly $h$ hops away from each other. Different from $C_{r, h}$, which marks the area where the sensor nodes are $h$ hops far away from the sensor nodes within $\Theta$, $E^{(h)}_{r, h}$ indicates the set of sensor nodes which are $h$ hops neighbours of the sensor nodes in $\Theta$. Note also that $E^{(h)}_{r, h} = \{s \mid s \in C^{(h)}_{r, h}\}$.

An illustrative example of the notations $E^{(h)}_{r, h}$ and $E^{(h)}_{r, h}$ is shown in Figure 6. In this example, four sensor nodes, $s_1, s_2, s_3, s_4$, having the same communication range, $r$, are considered. Given that $\Theta = \{s_1\}$, $C^{(1)}_{r, h}$ is the circle...
centred at $s_1$ with radius $r$, as shown in the shadow area of Figure 6a. $C_{r,\Theta}^{(2)}$ is the crescent-like area encompassed by the set of points \{a,b,c,d,e,f\}, as shown in the shadow area of Figure 6b. Similarly, $C_{r,\Theta}^{(3)}$ and $C_{r,\Theta}^{(4)}$ are the areas encompassed by the set of points \{a,f,e,g,h\} and the set of points \{e,d,g\}, as respectively, shown in the shadow areas of Figures 6c and 6d. Correspondingly, $E_{r,\Theta}^{(1)} = \{s_1\}$, $E_{r,\Theta}^{(2)} = \{s_2\}$, and $E_{r,\Theta}^{(3)} = \{s_4\}$.

**Figure 6** Illustrative examples for the notations $C_{r,\Theta}^{(1)}$ and $E_{r,\Theta}^{(i)}$:
(a) $C_{r,\Theta}^{(1)}$; (b) $C_{r,\Theta}^{(2)}$; (c) $C_{r,\Theta}^{(3)}$; and (d) $C_{r,\Theta}^{(4)}$ (see online version for colours)

Let $s_1$ be within the deployment group $G_{i,j}$ with the deployment point $(x_i, y_j)$ and let $\Theta = \{s_1\}$ initially. Given a pair of sensor nodes $s_1$ and $s_2$, the probability that the number of hops required for sending messages from $s_1$ to $s_2$ is $h$, is:

$$p_h = \int_{C_{r,\Theta}^{(h)}} \frac{1}{2\pi\sigma} e^{-\frac{(x-x')^2+(y-y')^2}{2\sigma^2}} \, dx \, dy. \quad (19)$$

Finally, after $p_h$’s are calculated, the expected communication overhead $E[X]$ within the same group can be calculated.

In addition, we investigate the communication overhead between two different deployment groups. Figure 7 shows an example of message delivery between two horizontally or vertically adjacent deployment groups. In this example, sensor nodes, $s_{n_1}$ and $s_{n_2}$, respectively, belong to two different deployment groups $G_{2,3}$ and $G_{2,4}$, and can securely communicate with each other with one hop transmission. Thus, when $s_1$ wants to send a message to $s_2$, the message is routed from $s_1$ to $s_2$ via the intermediaries $s_{n_1}$ and $s_{n_2}$. As a result, the expected communication overhead between two deployment groups, $G_{1/2}$ and $G_{3/4}$, is at most $\left(\lfloor i_2 - i_1 \rfloor + |j_2 - j_1| + 1\right) \cdot E[X] + (|i_2 - i_1| + |j_2 - j_1|)$.

One point needs to be further clarified in the above derivation, i.e. whether there exists at least one pair of sensor nodes that are from different deployment groups having the properties that their communication ranges cover each other and key sharing exists between them, i.e. a secure link exists between the two nodes coming from different deployment groups. Here, we define the term bridge as a secure link in which the two nodes are from two different deployment groups. For a deployment group $G_{i,j}$, since the probability that the distance between the resident point and deployment point is less than $3\sigma$, where $\sigma$ being the standard deviation is 0.9987, we simply assume that the distance between the deployment point and any resident point of $G_{i,j}$ is at most $3\sigma$. Due to the restriction of communication range, even if such pairs exist, they only exist in the overlapped region of two deployment groups. Consider two horizontally or vertically adjacent deployment groups, $G_{i,j}$ and $G_{r,j'}$. Let $D_{i,j} = (x_i, y_j)$ and $D_{r,j'} = (x_r, y_{j'})$, where $|x_i - x_r| + |y_j - y_{j'}| = 1$, be two deployment points from two different but vertically or horizontally adjacent deployment groups. Let $W = C_{3\sigma, D_{i,j}} \cap C_{3\sigma, D_{r,j'}}$ be the intersection of two circles of radius $3\sigma$, respectively, centred at $D_{i,j}$ and $D_{r,j'}$. The probability $H_i(W)$ that sensor nodes from $G_{r,j'}$ are located within $W$ is:

$$H_i(W) = \int_W \frac{1}{2\pi\sigma} e^{-\frac{(x-x')^2+(y-y')^2}{2\sigma^2}} \, dx \, dy. \quad (20)$$

Let $S_W$ be the set of sensor nodes from $G_{i,j'}$ located within $W$. Then, the probability that no sensor node from $G_{i,j}$ is also the neighbour of the sensor nodes from $S_W$ is:

$$H_j(W) = \left(1 - \int_{C_{r,s_W}} \frac{1}{2\pi\sigma} e^{-\frac{(x-x')^2+(y-y')^2}{2\sigma^2}} \, dx \, dy \right)^n, \quad (21)$$

where $n$ is the number of sensor nodes in each deployment group.
Furthermore, the probability $H_2(\mathcal{W})$ that there exists at least one sensor node in $G_{ij}$, which is also the neighbour of the sensor nodes in $G_{ij'}$, is defined as:

$$H_2(\mathcal{W}) = 1 - H_1(\mathcal{W}).$$

Eventually, given two vertically or horizontally adjacent deployment groups, the probability that there exists at least one bridge between them is:

$$H_1(\mathcal{W}) \cdot H_2(\mathcal{W}).$$

An example illustrating the number of bridges between two horizontally or vertically adjacent deployment groups is shown in Figure 8. It shows that the number of bridges is large enough to handle communication between the sensor nodes from two deployment groups.

**Figure 8** Relationship between average number of bridges between two deployment groups and number of sensors in a deployment group. The upper curve shows the number of bridges between two horizontally/vertically adjacent deployment groups $G_{ij}$ and $G_{ij'}$ with $d_{ij} = 30$ and $d_{ij'} = 20$, respectively. The lower curve shows the number of bridges between two horizontally/vertically adjacent deployment groups $G_{ij}$ and $G_{ij'}$ with $d_{ij} = 30$ and $d_{ij'} = 10$, respectively (see online version for colours).

4.4 The impact of the data sink on network integrity

Certainly, eavesdropping is one common means for adversaries to obtain the messages transmitted over WSNs. In addition to eavesdropping, in this work we consider that adversaries can launch node capture attacks, which can be divided into two kinds based on the actions of adversaries. The first one is passive node capture, which means that after physically capturing a sensor node adversaries can only steal the data stored in the sensor node. The second one is active node capture, based on which adversaries not only steal the data stored in the sensor nodes but also put the captured sensor nodes back into the WSNs as agents for further collection of messages.

In our method, for a pair of sensor nodes in WSNs, a pairwise key is used. It is widely believed that WSNs with pairwise key distribution have perfect resilience against node capture (Chan et al., 2003). In other words, the fraction of total communication compromised is almost linearly proportional to the number of sensor nodes compromised if adversaries launch passive node capture attack. However, when adversaries launch active node capture attack, the captured sensor nodes can be put back into WSNs to listen and steal the traffic through the captured sensor nodes. As a pair of sensor nodes that want to share a message might be connected by a single secure path, the capture of sensor nodes over this path also compromises the secure communication paths involving these two sensor nodes. Hence, as the number of sensor nodes captured increases, the fraction of total communication compromised could be increased beyond being linearly proportional to the number of captured sensor nodes as when active node capture attack is encountered.

An example illustrating the case of key sharing when sensor nodes are deployed to form a single deployment group is shown in Figure 9. In this example, the number of sensor nodes is 10. Provided that sensor nodes 1, 6, 7 and 10 are captured by adversaries, the secure link, such as the link between sensor nodes 8 and 9, remains secure. However, as the secure path between, for example, sensor nodes 2 and 8 is composed of at least one of the captured sensor nodes, the communication between nodes 2 and 8 is no longer secure.

**Figure 9** An example of key sharing in a single deployment group: (a) a sensor network with 10 sensor nodes; (b) blue nodes (i.e. nodes 1, 6, 7 and 10) denote the sensor nodes being captured; and (c) after some sensor nodes are captured, the network is divided into two parts. Both parts work well but they cannot communicate with each other. In this case, the size of the component containing the data sink (i.e. node 2) is 4 (i.e. nodes 2, 3, 4 and 5) (see online version for colours).

On the other hand, to query the status of the region controlled by the WSN, the data sink is placed at a specified location and the data collected by sensor nodes are aggregated to the data sink. However, the problem of resistance to attacks becomes more delicate when the location of a data sink is taken into consideration. Since a node capture attack will most likely break a WSN into two parts, this means that these two parts cannot communicate with each other; henceforth, the network’s integrity is said to be broken. As shown in Figure 9, if the data sink is sensor node 2 and the sensor nodes 1, 6, 7 and 10 are captured, the remaining network can be seen as a network composed of only sensor nodes 2, 3, 4 and 5 because, even if sensor nodes 8 and 9 are able to
securely communicate with each other and have enough battery power to sense the region, they cannot aggregate the sensed data to the data sink. The reason we claim that the problem becomes more delicate is based on the observation that, if the captured sensor nodes are the same but the data sink is different, then the surviving network will differ accordingly. Therefore, maintaining the integrity of the network is an important issue. In the next section, we describe how the integrity of the network can be preserved from the viewpoint of resilience against node capture.

4.5 Resilience against node capture

As mentioned above, when the Kryptograph is divided into at least two connected components, the fraction of total communication compromised will be increased abruptly and significantly. Since the minimum number of edges required for separating a regular graph \( R_{n,d} \) into two connected components is at least \( d \), the minimal number of sensor nodes captured by adversaries in order to partition the graph is the degree of each sensor node in the deployment group \( G_{i,j} \),

\[
\min_{A} A_{id} + \sum_{j \neq i} \delta_{A,j} \\text{subject to} \\sum_{j \neq i} \delta_{A,j} \leq n \]

where \( A_{id} \) contributes to the communication between deployment groups, the minimal number of sensor nodes required to be captured in order to partition a graph by isolating a deployment group is upper bounded by \( n \sum_{j \neq i} \delta_{A,j} \), where \( n \) is the number of sensor nodes in a deployment group.

5 Simulations

In this section, system configuration is first described in Section 5.1. Then, the parameters used in the key pre-distribution phase are empirically determined and discussed in Section 5.2. Finally, simulation results are provided in Section 5.3 to verify our analytic results.

5.1 System configuration

In the simulation, the following setting was used in our method when comparing our method with the methods of Chan et al. (2006) and Du et al. (2004). The deployment groups were arranged into a 5 \times 5 grid, i.e. there were 25 deployment groups. The number of sensor nodes in a deployment group was 100. Therefore, there were in total 2500 sensor nodes, whose distribution is assumed to be a two-dimensional Gaussian distribution. The distance between each group was 100 m. Thus, the base coefficients for deployment groups formed a 5 \times 5 matrix

\[
\begin{bmatrix}
1 & 1 & 2 & 2 & 1 \\
1 & 3 & 2 & 1 & 1 \\
2 & 2 & 1 & 1 & 1 \\
2 & 1 & 1 & 5 & 5 \\
2 & 1 & 1 & 5 & 1 \\
\end{bmatrix}
\]

The descendent factor \( \alpha \) was set to 2 and attack influence factors \( \{g_1, g_2\} \) were set to \{0.3, 0.03\}. In addition, the base coefficient for data sink for each deployment group was also a 5 \times 5 matrix

\[
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 5 \\
\end{bmatrix}
\]

where the descendent factor for data sink \( \tilde{\alpha} \) was set to 1 and attack influence factor \( \{g_1\} \) was set to \{0.5\}.

In addition, the setting of \( D_i(\cdot) \), based on the difference between the maximum attack probability and minimum attack probability of all deployment groups, was divided into five parts with four (i.e. \( \Omega = 4 \)) thresholds, \( \pi_1 = 20, \pi_2 = 26, \pi_3 = 38, \pi_4 = 42 \), and \( \pi_5 = 62 \). On the other hand, for \( D_i(\cdot) \), \( \phi_1 = 0.5, \phi_2 = 0.2 \), and \( \phi_3 = 0 \) were set. A guideline of selecting these parameters will be empirically revealed in Section 5.2.

In Chan et al.’s (2006) method, the probability of key sharing for a pair of sensor nodes in a group was set to 0.3. In Du et al.’s (2004) method, the horizontal or vertical overlapping factor was set to 0.2, and the diagonal overlapping factor was set to 0.05. For these two methods, the size of key pool was 25,000. In Chan et al.’s (2006) and Du et al.’s (2004) methods, the attack probability of each group was the same as in our method. For all the three methods, the communication range, \( r \), was set to 40 m and the variance of \( r \) was set to 1600 m².

5.2 Impact of parameters

In Section 3.3, the intra-group degree determination function [equation (7)] and inter-group degree determination function [equation (8)] are introduced in the key pre-distribution phase. They are related to some parameters. In this section, we will discuss how these parameters can be empirically determined via simulations.

For the intra-group degree determination function, \( D_i(\cdot) \), \( \Omega \) thresholds determine \( \Omega + 1 \) degree values according to equation (7). By varying \( \Omega \) values, two metrics will be examined to show how \( \Omega \) can be selected. First, the relationships between the size of the component containing data sink and the number of captured sensors under different settings of average memory overhead of a sensor node [equation (9)] are shown in Figures 10 and 11. In Figures 10 and 11, it can be observed, that although the increase of \( \Omega \) implies higher network integrity (corresponding to larger size of a network component containing the data sink), our simulation results show that this phenomena appears saturated when \( \Omega \simeq 2 \). Second, the relationship between the fraction of compromised communication and the number of captured sensors under different settings of average memory overhead is shown in Figures 12 and 13. We can observe
from Figures 12 and 13 that the results show similar tendency. By integrating the above evaluation results, it is sufficient to set $\Omega \geq 2$ according to our simulations.

**Figure 10** Relationships between the size of component containing data sink and number of captured sensors under the setting of $\Omega$ for average memory overhead, 90 (see online version for colours)

For the inter-group degree determination function, $D_{c_i,j}(p_{i,j}, p_{r,j'})$, it depends on three parameters, $\phi_1$, $\phi_2$, and $\phi_3$, which are determined according to the relationship between $G_{i,j}$ and $G_{r,j'}$ [equation (8)]. Based on the standard group-based deployment model (Du et al., 2004), it is reasonable to assume that $\phi_3 = 0$ for $G_{i,j}$ and $G_{r,j'}$ that are non-adjacent. Therefore, we only derive the values of $\phi_1$ and $\phi_2$ under the condition that the average memory overhead [see equation (9)] of a sensor node is a constant.

**Figure 11** Relationships between the size of component containing data sink and number of captured sensors under the setting of $\Omega$ for average memory overhead, 125 (see online version for colours)

By varying values of $\phi_1$ and $\phi_2$, two metrics will be examined to show how $\phi_1$ and $\phi_2$ can be selected. First, we examine the relationship between the size of the component containing data sink and the number of captured sensors under different settings of $\phi_1$ and $\phi_2$, as shown in Figures 14 and 15. It can be observed from Figures 14 and 15 that the differences among the results obtained from different settings of $\phi_1$ and $\phi_2$ are insignificant. Second, we also examine the relationships between the fraction of compromised communication and the number of captured sensors under different settings of $\phi_1$ and $\phi_2$, as shown in Figures 16 and 17. It is obvious in observation that there is no significant difference between the results obtained from...
different settings. In summary, these two evaluation results suggest flexibility in selecting these two parameters as long as they satisfy the constraint of $\phi_1 > \phi_2$, as having been described in equation (8).

**Figure 14** Relationships between the size of component containing data sink and number of captured sensors under the setting of $\phi_1, \phi_2$ for average memory overhead, 90 (see online version for colours)

**Figure 15** Relationships between the size of component containing data sink and number of captured sensors under the setting of $\phi_1, \phi_2$ for average memory overhead, 125 (see online version for colours)

5.3 Numerical results

In this section, the numerical results obtained from our method, Du et al.’s (2004) method, and Chan et al.’s (2006) method are presented and compared. We will discuss the following issues: (a) comparison of memory overhead, local connectivity, and global connectivity; (b) the relationship between the size of component consisting of data sink and number of captured sensors; (c) the relationship between the fraction of compromised communication and number of captured sensors; and (d) the relationship between the expected communication overhead and number of captured sensors.

**Figure 16** Relationships between the fraction of compromised communication and number of captured sensors under the setting of $\phi_1, \phi_2$ for average memory overhead, 90 (see online version for colours)

**Figure 17** Relationships between the fraction of compromised communication and number of captured sensors under the setting of $\phi_1, \phi_2$ for average memory overhead, 125 (see online version for colours)

5.3.1 Comparisons among memory overhead, local connectivity, and global connectivity

A comprehensive comparison of memory overhead, local connectivity, and global connectivity for our proposed method is shown in Table 2. It is observed that both the local connectivity and global connectivity strictly increase when memory overhead increases.

The local connectivity in the proposed method is inferior to that proposed by Du et al. (2004) due to the following reasons. In Du et al. (2004), the memory overhead is the same for each deployment group. However, in the proposed method, the degree of key sharing is not the same for each deployment group but depends on the attack probability of each deployment group. Hence, the sensors in the deployment group with lower/higher attack probability have lower/higher level of key sharing. In addition, due to the limited communication range, the local connectivity in the deployment group with higher
an application-driven attack only slightly increases. On the other hand, the decrease of key sharing intensively affects the local connectivity in the deployment group with lower attack probability. As a whole, although the local connectivity of our method is inferior to the method of Du et al. (2004), the global connectivity of our method is similar to that of Du et al. (2004). However, it should be noted that the connectedness of the network is primarily determined by global connectivity and is not determined by local connectivity.

Table 2  A comprehensive comparison of memory overhead, local connectivity, and global connectivity for our method

<table>
<thead>
<tr>
<th>Memory usage</th>
<th>Local connectivity</th>
<th>Global connectivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.09</td>
<td>0.80</td>
</tr>
<tr>
<td>70</td>
<td>0.13</td>
<td>0.95</td>
</tr>
<tr>
<td>96</td>
<td>0.19</td>
<td>0.98</td>
</tr>
<tr>
<td>120</td>
<td>0.24</td>
<td>0.99</td>
</tr>
<tr>
<td>160</td>
<td>0.32</td>
<td>0.99</td>
</tr>
<tr>
<td>200</td>
<td>0.40</td>
<td>1.00</td>
</tr>
</tbody>
</table>

As for Chan et al.’s (2006) method, the authors presented an idea similar to ours in that deployment groups with different attack probabilities should be associated with different degrees of connectivity in key sharing. However, they did not take both local and global connectivity into consideration. In their method, each group does not have its own key pool; instead, each node chooses its own key ring from a single but big key pool. Under this circumstance, one sensor node may share keys with another one that is far from it. Therefore, the local and global connectivities in Chan et al. (2006) are very poor when the communication range of each sensor node is considered.

5.3.2 Relationship between the size of component consisting of data sink and number of captured sensors

After certain sensor nodes are captured, the Kryptograph is perhaps divided into many connected components, which cannot securely communicate each other. Since the network we can control is only the component containing a data sink, it will reduce the scale of the network if this situation happens. As shown in Figure 18, as the number of captured sensor nodes becomes large, the size of the component consisting of data sink is gradually reduced. It can be observed from Figure 18 that our method is superior to Du et al.’s (2004) method, which did not consider the attack probabilities of deployment groups and is superior to Chan et al.’s (2006) method, which did not consider the communication range of each sensor node. Since the number of total sensor nodes is 2500, the component containing the data sink in our method is still the maximum one in the whole network after some sensor nodes are captured by adversaries. In Du et al. (2004) and Chan et al. (2006), when adversaries capture more than 800 nodes, the size of a component containing a data sink is almost equal to 1 leading to the network crashing completely.

Figure 18  After some sensor nodes are captured, the size of the component consisting of data sink is decreased. In this figure, \( m \) denotes the number of keys stored in a sensor and \( m^* \) denotes the average number of keys stored in a sensor (see online version for colours)

5.3.3 Relationship between the fraction of compromised communication and number of captured sensors

The relationship between the fraction of communication compromised and the number of sensor nodes captured is shown in Figure 19. Since the capture of sensor nodes will incur the compromise of some other communication links in the probabilistic key pre-distribution and will not incur the compromise of other communication links in the deterministic key pre-distribution, under the situation where the number of sensor nodes captured is the same, the fraction of communication compromised in deterministic key pre-distribution is less than that in the probabilistic key pre-distribution.

Figure 19  Relationship between the number of captured sensor nodes and the fraction of compromised communication. In this figure, \( m \) denotes the number of keys stored in a sensor and \( m^* \) denotes the average number of keys stored in a sensor (see online version for colours)
5.3.4 Relationship between the expected communication overhead and number of captured sensors

The relationship between the expected communication overhead and the number of sensor nodes captured is shown in Figure 20. We do not compare our results with the expected communication overhead in methods of Du et al. (2004) and Chan et al. (2006) because the networks constructed by their methods will be divided into many parts due to node capture attack such that the size of the components generated by our method and their method are completely different. Certainly, the larger the component, the larger the expected communication overhead. Thus, it is not fair to compare the expected communication overhead in two network components with different sizes. When the number of sensor nodes captured is increased, usually the message delivery between the source sensor node and target sensor node must detour around the communication compromised rather than on the original secure path. Hence, such detours will increase the communication overhead between any pair of sensor nodes.

Figure 20 The example shows the situations about the average number of communication overhead of network. In this figure, \( m \) denotes the number of keys stored in a sensor and \( m^* \) denotes the average number of keys stored in a sensor.

![Graph showing the relationship between the expected communication overhead and number of captured sensors](image)

6 Related work

6.1 Deterministic key pre-distribution methods

In the literature, there have been some deterministic key pre-distribution methods proposed for WSNs. The first deterministic key pre-distribution protocol, called random pairwise keys pre-distribution, was proposed by Chan et al. (2003). Based on an idea similar to Chan et al. (2003), a promising key establishment scheme called Peer Intermediate for Key Establishment (PIKE) was proposed by Chan and Perrig (2005). In PIKE, the key pre-distribution model is a grid structure and each sensor node is given a 2-D label according to its location in the grid. For key establishment, two sensor nodes will share a unique key if they are located at either the same row or column. Thus, in PIKE each sensor node stores \( 2(\sqrt{N} - 1) \) keys, where \( N \) denotes the total number of sensor nodes. Çamtepe et al. (2006) proposed an expander graph based deterministic key pre-distribution scheme, which differs from PIKE merely in the structure of key sharing. Basically, the key graph in Çamtepe et al. (2006) is an expander graph while that in PIKE (Chan and Perrig, 2005) is a grid. Theoretically, the connectivity of an expander graph is better than that of a grid. The main disadvantage of deterministic key distribution methods is that two physically neighbouring sensor nodes may have the possibility that they do not have the shared key so that additional communication overhead is required to establish the secure link between them.

6.2 Location-aware key pre-distribution methods

The aforementioned key pre-distribution schemes only consider uniform deployment of sensor nodes, which means that all sensor nodes are constrained to being uniformly distributed over the entire sensing region. For different vehicles employed for deploying WSNs, the resultant deployment patterns of sensor nodes are different (Du et al., 2004). By taking deployment pattern in consideration, key management schemes using deployment knowledge to improve key usage and enhance the resistance to node capture were proposed by Du et al. (2004) and Liu and Ning (2005). Liu et al. (2005a) proposed a general framework for key pre-distribution based on group-based deployment. Based on simply partitioning the sensing region and using the error correction code, Huang and Medhi (2007) established a secure pairwise key pre-distribution approach. However, all probabilistic key pre-distribution methods suffer from the same predicament: when the number of captured sensor nodes increases, the fraction of compromised communication increases accordingly. In view of this, Zhou et al. (2005) proposed a pairwise key pre-distribution scheme for group-based deployment at the expense of encountering higher memory overhead. Also based on analysing deployment information, Liu and Ning (2003a) proposed a key pre-distribution protocol to reduce the memory overhead and mitigate the impact of node capture attack, while Huang et al. (2004) proposed a key pre-distribution protocol by providing analyses for selective node capture attack and node fabrication attack.

6.3 Other key pre-distribution methods

In addition to the deterministic key pre-distribution methods and location-aware key pre-distribution methods described above, there are other types of security protocols deserve to be described here. Motivated by Blom (1984) and Blundo et al. (1992), Liu and Ning (2003b) and Du et al. (2003), respectively, proposed to use bivariate polynomials and rows of symmetric matrices as the keys of the key pool to improve resistance to node capture attack. Delgosha and Fekri (2005) further utilised the similar idea of Liu and Ning (2003b) to develop a multivariate-based key pre-distribution method. Zhang et al. (2007) refined Blundo et al.’s (1992) method by utilising random perturbation to establish the pairwise keys between every pair of sensor nodes. Based on
similar idea of Zhang et al. (2007) but with constrained random perturbation, Yu et al. (2010) refined Blom’s (1984) method to derive a non-interactive key distribution scheme for WSNs. On the other hand, multiple path technique was also exploited by Li et al. (2005) and Wacker et al. (2005) to enhance the security of key pre-distribution protocols. Conti et al. (2007) proposed a method in which a pair of sensor nodes can collaboratively negotiate a common key with the help of the other sensor nodes. Moreover, when considering a heterogeneous sensor network, Traynor et al. (2006) analysed the corresponding performance of key distribution methods. Zhu et al. (2003) proposed a key management protocol in which a family of pseudo-random functions are used to authenticate node and derive pairwise keys. Finally, a Routing-Driven Elliptic Curve Cryptography Based Key Management Scheme for Heterogeneous Sensor Networks has been recently introduced by Du et al. (2009).

7 Conclusions

In this paper, an attack probability-based deterministic key pre-distribution mechanism for secure communication in wireless sensor networks has been presented. We first study a new model of calculating the attack probability of each deployment group. Then, a new deterministic key pre-distribution mechanism for group-based deployment by taking attack probability into consideration is proposed. We further investigate several issues, including communication overhead, memory overhead, and resistance to node capture attack to verify the suitability of the proposed method. The key idea is to consider the location of data sink in order to help the attack probability determination of a deployment group. Simulation comparisons with some known methods further demonstrate the superiority of our method.

Acknowledgements

This research was supported by NSC 97-2221-E-001-008, NSC 98-2221-E-001-004-MY3, and NSC 97-2221-E-002-216-MY3.

References


Notes
1 The terms ‘edge’ and ‘link’ are used interchangeably.
2 Note that although it has been proved by Guan and Yu (2005) that key pre-distribution schemes achieve better performance when the sensing region is divided into hexagons, our proposed method with minor modifications can also be applied to the hexagon regions.
3 If for some reason, such as those sensor nodes being protected by tamper resistance devices, adversaries cannot directly capture all of the sensor nodes neighbouring the data sink, the adversaries still can capture the sensor nodes near those protected by tamper resistance devices to also gain high payoff, although this pay-off is lower than the previous one.
4 Consider two one-dimensional identical Gaussian distributions lying side-by-side. Assume that they overlap with each other. Now, fix the location of one distribution, and move the location of another distribution. Once their variances are known, the distribution can be moved to a proper location so that the probability of their overlap is as nearly large as their mean values. As a result, when these two distributions are jointly considered, they behave like uniformly distributed. In other words, though such movement of the distribution does not result in the exact uniform distribution. This idea can be generalised to the case of multiple two-dimensional Gaussian distributions.