Combinatorial design-based key pre-distribution schemes for wireless sensor networks

Chanjun Yang

Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Recommended Citation

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Combinatorial design-based key pre-distribution schemes for wireless sensor networks

by

Chanjun Yang

A thesis submitted to the graduate faculty in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Johnny S. Wong, Major Professor
Wallapak Tavanapong
Richard Ng

Iowa State University
Ames, Iowa
2006

Copyright © Chanjun Yang, 2006. All rights reserved.
Graduate College
Iowa State University

This is to certify that the master's thesis of

Chanjun Yang

has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy
DEDICATION

This thesis is dedicated to my Mom and Dad, without whom I would not have been able to complete this work.
TABLE OF CONTENTS

LIST OF TABLES ...................................................... vi
LIST OF FIGURES .................................................... vii
ABSTRACT .............................................................. viii
CHAPTER 1. Introduction and motivation ............................. 1
CHAPTER 2. Related work and background ............................ 4
  2.1 Related work .................................................... 4
     2.1.1 Probabilistic schemes ...................................... 4
     2.1.2 Deterministic schemes ...................................... 5
     2.1.3 Hybrid schemes ............................................. 5
  2.2 Combinatorial design and finite projective plane ............... 6
CHAPTER 3. Combinatorial design-based key management schemes 9
  3.1 Random key pre-distribution scheme .......................... 9
  3.2 Basic scheme: FPP-based symmetric scheme .................. 10
  3.3 FPP-complementary hybrid scheme .......................... 12
  3.4 FPP-random hybrid scheme ................................. 12
  3.5 FPP-based two-party scheme ................................. 13
     3.5.1 Basic idea ............................................... 14
     3.5.2 Detailed description ...................................... 14
CHAPTER 4. Performance evaluation .......................... 17

4.1 Performance metrics ............................................. 17
4.2 Analytical result .................................................... 18
4.2.1 Key connectivity ................................................. 18
4.2.2 Scalability ....................................................... 23
4.2.3 Resilience against node capture ......................... 25
4.2.4 Communication overhead ............................... 29
4.2.5 Computational complexity ............................ 30

4.3 Performance analysis ............................................ 30
4.3.1 Comparison of schemes ............................... 30
4.3.2 Experimental results ........................................... 35

CHAPTER 5. Conclusion and future work ......................... 43

BIBLIOGRAPHY ......................................................... 45

ACKNOWLEDGEMENTS .................................................. 47
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 4.1</td>
<td>Key connectivity ($N = 10,000, K = 72$)</td>
<td>32</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>Resilience against node capture ($N = 10,000, K = 72$)</td>
<td>33</td>
</tr>
<tr>
<td>Table 4.3</td>
<td>Resilience against node capture ($N = 10,000, K = 72, C_1 = 0.6$)</td>
<td>34</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

| Figure 4.1 | Key connectivity                                      | 38 |
| Figure 4.2 | Resilience against node capture when size of key pools changes | 38 |
| Figure 4.3 | Resilience against node capture with the changes of the number of nodes captured | 40 |
| Figure 4.4 | Key connectivity with the change of the network size and key ring size | 40 |
| Figure 4.5 | Key connectivity with the change of the network size | 41 |
| Figure 4.6 | Resilience against node capture with the change of the network size | 41 |
ABSTRACT

Pair-wise key establishment between neighboring nodes in wireless sensor networks is an essential security service since sensor networks are usually deployed in hostile environments and under various attacks. However, techniques that are used in general networks are unsuitable for sensor networks because of the resource constraints of the sensor nodes. Recently, key pre-distribution techniques have been proposed to establish pair-wise keys. However, since the network topology information is not available before deployment, it is difficult to decide which keys to preload to individual sensor node. Several types of schemes have been proposed to deal with this problem. Probabilistic schemes randomly pick some keys from a large key pool to form a key ring for each node. Deterministic schemes construct the key ring of each node based on some kind of rules (e.g. combinatorial design). There are also some hybrid schemes that combine the probabilistic and deterministic designs. In this paper, we propose two novel schemes based on the combinatorial design. Detailed analytical and experimental results are given to compare the performance of the schemes discussed in this paper, which also indicate that our schemes achieve a better key connectivity while still maintain a good resilience against node capture.
CHAPTER 1. Introduction and motivation

Large-scale wireless sensor networks (WSN) are becoming more and more popular recently due to the advances in wireless communication and electronics [Akyildiz et al. (2002)]. Sensor networks can be used for a lot of applications, such as environmental observation, building safety monitoring, military sensing and tracking, health care, etc. Typical sensor nodes are low-cost, low-power and limited in storage, communication and computation capabilities. For example, a widely used research platform is Berkeley Mica Motes, which uses Atmel Atmega 128L processor (4MHZ) with 128KB on-board flash memory and 4K bytes SRAM [Berkeley (2002)]. The sensors have a radio range of 100 feet and can communicate at a rate of 40Kbits/sec. They use 2AA batteries as power supply. Sensor networks typically consist of up to tens of thousands resource constrained sensors, which sense the local information and communicate with neighboring nodes to send data back to the base station.

Since many sensor networks are deployed in unattended or hostile environments, they are prone to various kinds of attacks. Thus security services become critical in sensor networks. Basic security services include key distribution and management. In a lot of applications, most communications are between a sensor and its immediate neighbors. Therefore, we require each pair of neighboring nodes in a WSN to establish a pair-wise key for secure communication. However, the properties of large scale WSN make the pair-wise key distribution and management problem much more challenging. It is infeasible to apply most of the traditional techniques that are widely used in general network environments. For example, one of the key management schemes is to use a key
distribution center (KDC) to distribute all the keys. This is not desirable for WSN because sensors are usually deployed in hostile environment and no trusted infrastructure is available. Besides, using KDC adds a lot of communication overhead to WSN, which is not desired because of the low power supply of sensor nodes. Another approach is to use public key cryptography. However, sensor nodes are low-power devices with limited computation capability, which makes it unsuitable for public key algorithms with high processing requirements. Therefore, the pre-distribution key management schemes are proposed, where keys are preloaded in the sensors before deployment. These approaches eliminate the costly communication and computation required for distributing keys after sensors are deployed. But the random deployment pattern of WSN introduces other difficulties for the key pre-distribution schemes. Due to the hostile environment and large size of WSN, manual deployment of sensors is almost impossible in many applications. Sensors are usually randomly scattered from airplanes to the deployment areas. Therefore, the exact position of a sensor is unknown before deployment, which implies that a sensor cannot know its neighborhood a priori. The unknown network topology makes it hard to decide which encryption keys to pre-load in each node before deployment.

In this thesis, we propose two new key pre-distribution techniques based on a combinatorial design: Finite Projective Plane (FPP). The first one is based on the combination of the basic FPP-based scheme and the probabilistic scheme while the second one is to use two FPPs instead of one for the design. The motivations of our designs are to deal with the scalability problems of the basic FPP-based scheme and to improve resilience against node capture at the same time. However, there is no scheme that is suitable for all applications. Mathematical analysis and experimental results are provided to compare the performance of various schemes, including our proposed ones.

The rest of the thesis is organized as follows: chapter 2 gives a brief overview of the previous approaches proposed and introduces the background knowledge of combinatorial design and finite projective plane. chapter 3 first describes three existing key
pre-distribution schemes, including the random scheme [Eschenauer and Gligor (2002)], FPP-based symmetric scheme [Camtepe and Yener (2004); Lee and Stinson (2005)], and the hybrid scheme [Camtepe and Yener (2004)]. Then we present our two new schemes: FPP-random hybrid scheme and FPP-based two-party scheme. Performance analysis of the five schemes is given in chapter 4, followed by the analytical and experimental results. Chapter 5 concludes the thesis and discusses potential future work.
CHAPTER 2. Related work and background

2.1 Related work

A number of key pre-distribution schemes have been proposed in the past several years to meet the security requirement of WSN. Most of them fall into three categories: probabilistic, deterministic, and hybrid schemes.

2.1.1 Probabilistic schemes

The basic probabilistic scheme was proposed in [Eschenauer and Gligor (2002)]. In this scheme, each node is preloaded with a random set of keys from a large key pool. After deployment, two neighboring nodes that share a common key can establish a session key. However, for large key pools, the probability that two nodes share a common key is small. For small key pools, if one node is captured and all its keys are revealed, a lot of other communications are compromised as well because there are a lot of overlapping keys among the nodes. To improve the resilience against node capture, Chan et al. proposed the q-composite scheme [Chan et al. (2003)], which requires at least $q$ keys in common for two neighboring nodes to establish a session key using these common keys. Therefore, the keys used to encrypt messages are different from the original keys and when nodes are captured, less links are compromised. However, on the other hand, this scheme also decreases the key sharing probability and requires nodes to store more keys in order to establish pair-wise keys.
2.1.2 Deterministic schemes

Several deterministic key pre-distribution schemes were proposed in [Camtepe and Yener (2004); Lee and Stinson (2005)] based on combinatorial design theory. The scheme proposed in [Camtepe and Yener (2004)] uses a symmetric design finite projective plane with parameters \((n^2+n+1, n+1, 1)\) to support a WSN of size \(N = n^2+n+1\). According to the design, every node needs to store \(K = n + 1\) keys and every pair of nodes have exactly one key in common. Therefore after deployment, every pair of neighboring nodes can find a common key as their pair-wise key. Similar idea was also proposed in [Lee and Stinson (2005)].

2.1.3 Hybrid schemes

Although the finite projective plane (FPP)-based symmetric scheme guarantees every pair of nodes to establish a key after deployment, it requires each node to store about \(\sqrt{N}\) keys, which is not very scalable as the network size \(N\) increases. In [Camtepe and Yener (2004)], the authors proposed a hybrid design of finite projective plane and its complementary design to address the limitations of the basic scheme.

Another type of deterministic solutions is the polynomial based key pre-distribution scheme proposed in [Blundo et al. (1993)]. Unlike the previous solutions which directly store keys, this scheme stores a polynomial share \(f(u, y)\) of a symmetric \(t\)-degree polynomial \(f(x, y)\) in a node with the ID \(u\). After deployment, if node \(u\) needs to establish a pair-wise key with node \(v\), it can evaluate its polynomial share \(f(u, y)\) at the point \(y = v\) and compute \(f(u, v)\). Similarly, node \(v\) calculates \(f(v, u)\). Since \(f(x, y)\) is symmetric, \(u\) and \(v\) can then use \(f(u, v)\) as their pair-wise key. This scheme has the property that if no more than \(t\) nodes are captured \((t\) is the degree of the polynomial), the network is secure. But if more than \(t\) nodes are captured, the whole network is compromised, where \(t\) is the degree of the polynomial. To further improve the resilience of the net-
work, polynomial pool based schemes were proposed in [Liu and Ning (2003); Du et al. (2005)], which are combinations of the polynomial based scheme and the probabilistic scheme. To achieve good resilience, these schemes need to store more keying materials (coefficients of polynomial shares), which may not be scalable for sensor nodes with limited storages. Besides, these schemes require more computations than the schemes that explicitly store keys.

In some other work, deployment knowledge are also used to improve the performance of the key pre-distribution schemes [Du et al. (2004)]. They assume that the location of the nodes after deployment can be estimated. Therefore every node knows who is more likely to be its neighbors before deployment and the keys stored in the nodes are based on this deployment information. However, there are a lot of WSN applications in which the estimated locations of the nodes are not available.

### 2.2 Combinatorial design and finite projective plane

Informally, combinatorial design is defined as a way of selecting subsets from a finite set in such a way that some specified conditions are satisfied [Wallis (1988)]. The subsets selected from the universal set is called blocks. If there is no structural ordering or pattern in the blocks (blocks are only sets), this design is called a block design [Wallis (1988)]. Pre-distributing keys into sensor nodes is similar to constructing a block design. The key pool can be mapped to a universal set of keys and the key rings are all blocks. Regular design is a widely studied class of block designs. A regular design has four parameters: $v$, $b$, $r$, and $k$, where $v$ is the number of elements in the universal set $S$ and $b$ is the number of blocks in the design. In a regular design, blocks have the same size $k$. In other words, each block is a $k$-subset of $S$. Besides, each element in $S$ belongs to exactly $r$ blocks [Wallis (1988)].

**Definition 1** A $(v, b, r, k, \lambda)$-balanced incomplete block design (BIBD) is a $(v, b, r, k)$-
regular design and each pair of elements occur together in exactly \( \lambda \) of the blocks [Wallis (1988); Anderson (1990)].

A \((v, b, r, k, \lambda)\)-BIBD has the following two properties.

1. \( bk = vr \),
2. \( r(k - 1) = \lambda(v - 1) \).

The two properties can be proved using the balanced properties of regular and BIBD design.

**Definition 2** A symmetric BIBD is a \((v, b, r, k, \lambda)\)-BIBD with \( r = k \) and \( v = b \).

Therefore, a \((v, b, r, k, \lambda)\)-symmetric BIBD can be written as \((v, k, \lambda)\). In such a symmetric BIBD, there are \( b = v \) blocks. Every element is in \( r = k \) blocks and every block contains \( k = r \) elements. Every pair of elements occurs in \( \lambda \) blocks and every pair of blocks intersects \( \lambda \) elements.

**Definition 3** A finite projective plane of order \( n \) is a \((n^2 + n + 1, n + 1, 1)\)-symmetric BIBD, where \( n \) is a prime power.

Finite projective plane (FPP) is a subset of symmetric BIBD. As we can see from the definition, in a FPP, the block size is roughly equal to the number of blocks. It is well known that FPP can be implemented using Mutually Orthogonal Latin Squares (MOLS) [Wallis (1988); Anderson (1990)] of finite field. Therefore, it is possible to apply FPP design to the key pre-distribution scheme.

[Camtepe and Yener (2004)] introduces the idea of complementary design for the hybrid key pre-distribution scheme. For each block \( B_i, i = 1, \cdots, k \) in the block design \( D = (v, k, \lambda) \), a complementary symmetric block \( \overline{B}_i \) is constructed as \( S - B_i \). The complementary design \( \overline{D} \) of design \( D \) contains all the complementary blocks. \( \overline{D} \) is also
a symmetric design with the parameters $(v, v-k, v-2k+\lambda)$ [Camtepe and Yener (2004)].

For example, a finite projective plane with parameters $(7, 3, 1)$ on the set $\{1, 2, 3, 4, 5, 6, 7\}$ is given as $\{1, 2, 4\}, \{2, 3, 5\}, \{3, 4, 6\}, \{4, 5, 7\}, \{5, 6, 1\}, \{6, 7, 2\}, \text{ and } \{7, 1, 3\}$. And the corresponding complementary design is $\{3, 5, 6, 7\}, \{1, 4, 6, 7\}, \{1, 2, 5, 7\}, \{1, 2, 3, 6\}, \{2, 3, 4, 7\}, \{1, 3, 4, 5\}, \text{ and } \{2, 4, 5, 6\}$.
CHAPTER 3. Combinatorial design-based key management schemes

In this section, we present several key pre-distribution schemes that are based on finite projective plane (FPP) design. Since our schemes are related to the random key pre-distribution scheme proposed in [Eschenauer and Gligor (2002)] and we will compare the performance of combinatorial design based schemes with it, we briefly introduce the random scheme first.

3.1 Random key pre-distribution scheme

The basic random key pre-distribution scheme is proposed in [Eschenauer and Gligor (2002)]. In the rest of the thesis, we refer to this scheme as the random scheme. In this scheme, all the keys stored in the nodes are randomly picked from a key pool $P$. The random scheme includes three phases as follows:

Key pre-distribution: The scheme randomly generates a large number of distinct cryptographic keys, which forms a key pool $P$. Each sensor node randomly picks $K$ keys from $P$ and stores them in its memory. These $K$ keys forms the node’s key ring.

Direct establishment: After deployment, each node discovers its neighbors and whether it shares common keys with them. Node $u$’s neighbors are all the nodes that are in $u$’s communication radius. $u$ can then use these common keys to communicate with some of its neighbors. In order to find out the direct keys, nodes can exchange IDs of the keys they have, or they can broadcast a list of challenges as mentioned in [Eschenauer and
Gligor (2002)] in order to secure the communication.

Path key establishment: If two neighboring nodes do not share a common key in their key rings, they need to find a secure path to set up a key between them. A path is secure if every link on it has already established a communication key. Therefore one node can generate a path key and send it to the other node securely.

The random scheme is easy to implement and its security performance is not bad. However, the key connectivity of the random scheme is usually not good. Key connectivity is the probability that two neighboring nodes can establish a direct key after deployment. If the key pool is very large while the key ring is relatively small, the key connectivity is low because the probability that two nodes have common key is small. Lower key connectivity requires more communication in the path key establishment phase and is also not very secure.

On the other hand, if we decrease the key pool size or increase the key ring size, the key connectivity will be increased. However, more communication links will use the same keys and when one node is captured, more links are compromised as well.

Most schemes proposed later also consist of these three phases. The differences among various schemes mostly lie in the key pre-distribution phase. For instance, the combinatorial design based schemes we propose in this thesis introduce structured key pool and key rings in the key pre-distribution phase to improve the performance.

3.2 Basic scheme: FPP-based symmetric scheme

Finite projective plane has some nice symmetric properties, especially that every two blocks share exactly one key. A natural idea is to map the finite projective plane to the key pre-distribution scheme in the sensor network. The basic FPP-based symmetric scheme was proposed in [Camtepe and Yener (2004)].

Similar to the random scheme, the FPP-based symmetric scheme also begins with
the key pre-distribution phase. For a sensor network of \( N \) nodes, the key pre-distribution phase works as follows:

1. Construct a finite projective plane with parameters \((n^2 + n + 1, n + 1, 1)\), where \( n \) is the smallest prime power such that \( n^2 + n + 1 \geq N \). Mutually Orthogonal Latin Squares (MOLS) can be used to implement finite projective planes [Camtepe and Yener (2004)].

2. Each element in the projective plane is associated with a distinct key. So the key pool contains \( n^2 + n + 1(\geq N) \) keys. Key pool size \( P = n^2 + n + 1 \).

3. Each block is associated with a key ring of a distinct sensor node. Since there are \( n^2 + n + 1(\geq N) \) blocks, there will be enough blocks for all the sensor nodes.

4. The key ring size \( K \) of each node is equal to the block size \( k = n + 1 \). So given the network size \( N \), we can compute the key ring size \( K \) of each node.

5. Since this is a finite projective plane, every pair of blocks intersects \( \lambda = 1 \) elements, which means that every pair of sensor nodes shares exactly one key.

After the nodes are deployed, they perform the direct key establishment phase. Note that the path key establishment is not necessary for this scheme since each pair of nodes can establish a key in the direct key establishment phase.

FPP-based symmetric scheme is a deterministic scheme compared to the random scheme which is probabilistic. It solves the low key connectivity problem perfectly since every pair of nodes share exact one key which can be used to secure their link if they are neighbors. However, this scheme raises other problems. One essential one is the scalability problem. For a network of size \( n_1^2 + n_1 + 1 \leq N \leq n_2^2 + n_2 + 1 \), where \( n_1 \) and \( n_2 \) are two adjacent prime power, the key ring size is \( K = n_2 + 1 \). For example, if the network size \( N = 10,000 \), we need the key ring size to be at least 102 to use this scheme, while in random scheme, key ring size can be smaller.
3.3 FPP-complementary hybrid scheme

[Camtepe and Yener (2004)] also proposed another more complex hybrid scheme to solve the scalability problem of the FPP-based symmetric scheme. We refer to this scheme as FPP-complementary hybrid scheme. First a core design is generated using finite projective plane. Based on the core design, a complementary design is generated. For each key ring $K_i$ in the core design, a complement block $\overline{K}_i = P - K_i$ is generated, where $P$ is the key pool. If there are $N$ sensor nodes and each can carry $K$ keys, the key pre-distribution phase is performed as follows. First, find the largest prime power $n$ such that $n + 1 \leq K$ and generate the core symmetric design $(n^2 + n + 1, n + 1, 1)$ based on $n$. The key rings generated by the core symmetric design covers $N' = n^2 + n + 1$ sensor nodes. Then $n + 1$ complement blocks are generated for each key ring in core design. For the other $N - N'$ nodes, randomly pick a $K$-subset $H_i$ from the complementary blocks as its key ring. The direct and path key establishment phases are similar to previous schemes.

The FPP-complementary hybrid scheme solves the scalability problem of the FPP-based symmetric scheme. Compared with the basic scheme, FPP-complementary hybrid scheme uses the same key pool size to support more sensor nodes. Obviously, more communication links use the same keys in this scheme. If some nodes are captured, many remaining links are compromised too.

3.4 FPP-random hybrid scheme

In order to solve the scalability problem of FPP-based symmetric scheme and to support larger network size, we propose FPP-random hybrid scheme, which combines the random scheme with the symmetric design to improve scalability. Different from the FPP-complementary hybrid scheme, the FPP-random scheme can use various key pool size for fixed network size and key ring size based on different applications. The key
pre-distribution phase of this hybrid scheme works as follows:

1. Given the maximum key ring size $K$, find the largest prime power $n$ such that $n < K$.

2. Randomly generate a key pool of size $P$, where $P \geq n^2 + n + 1$.

3. Randomly pick $N' = n^2 + n + 1$ keys from the key pool $P$ and generate a finite projective plane $(n^2 + n + 1, n + 1, 1)$ on these $N'$ keys. This symmetric design is called the core design.

4. Use the finite projective plane to support $N'$ nodes in the network. For each of the other $N - N'$ nodes, randomly pick $K$ keys from the whole key pool as its key ring.

Unlike the basic FPP-based symmetric design, the FPP-random hybrid design requires the nodes to perform both direct key establishment and path key establishment after deployment because it sacrifices the 100% key connectivity in the basic design to improve the scalability. This scheme also improves the resilience against node capture since the key pool $P$ can be larger than that of the basic scheme. In the direct key establishment phase, if two neighboring nodes’ key rings are from the core design, they can establish a pairwise key for sure. Otherwise, they need to check whether they share a common key. If the two nodes do not share a common key, they need to start the path key establishment phase to obtain a communication key. The direct and path key establishment phases in the FPP-random hybrid scheme are similar to those in the random scheme.

3.5 FPP-based two-party scheme

Although the FPP-random scheme we proposed solves the scalability problem of the FPP-based symmetric scheme, it introduces new problems. When the network size
increases but the key ring size remains the same, more nodes choose their key rings randomly from the key pool. Therefore, the key connectivity of the scheme decreases and is close to that of the random scheme. To further improve the performance of FPP-random hybrid scheme, we present the idea of a new FPP-based two-party key pre-distribution scheme. The idea is to improve the scalability while still maintain good key connectivity. This scheme also improves the resilience against node capture compared to the basic FPP-based symmetric scheme.

3.5.1 Basic idea

Similar to all the previously discussed schemes, our FPP-based two-party scheme also includes three phases: key pre-distribution, direct key establishment, and path key establishment. However, the key pre-distribution phase is different from the FPP-based symmetric and FPP-random schemes. Instead of constructing only one finite projective plane on the key pool, we divide the whole key pool into two small ones and construct a finite projective plane on top of each of the small key pools separately. The two small key pools can have some overlapping keys so as to increase the key connectivity. The number of overlapping keys is an adjustable parameter based on different applications. In the key pre-distribution phase, \( \frac{N}{2} \) of the nodes are preloaded with key rings from the first finite projective plane, while the rest of them store the key rings from the second finite projective plane. The direct and path key establishment phases are similar to those in the random scheme.

3.5.2 Detailed description

3.5.2.1 Key pre-distribution

In this phase, the offline key setup server construct the key rings for all the sensor nodes. For a sensor network of size \( N \), the algorithm is as follows:
1. Find the smallest integer $N'$ that is greater than $\frac{N}{2}$ and also satisfy the following condition: $N' = n^2 + n + 1$, while $n$ is a prime power.

2. Generate two random key pools $K_1$ and $K_2$, where $|K_1| = N'$ and $|K_2| = \lceil(1 - \rho)N'\rceil$, where $0 \leq \rho \leq 1$, and $\rho$ is called overlapping factor, which is used to denote the proportion of overlapping keys in the key pools. Randomly pick $\lceil\rho N'\rceil$ keys from key pool $K_1$ and distribute them in $K_2$. Therefore $|K_1| = |K_2| = N'$.

3. Construct two finite projective planes $P(K_1)$ and $P(K_2)$ on top of the key pools $K_1$ and $K_2$.

4. For the first $\lceil\frac{N}{2}\rceil$ nodes, randomly pick a block from $P(K_1)$ as its key ring. For the rest nodes, randomly pick a block from $P(K_2)$ as its key ring.

5. The unused blocks are saved for the sensors that will join the network later.

The overlapping factor $\rho$ must be chosen carefully so as to achieve good key connectivity while still maintain satisfactory resilience against node capture.

3.5.2.2 Direct key establishment

In the direct key establish phase, each node discovers all its neighbors and whether it shares keys with them. For a pair of neighboring nodes, if their key rings are from the same finite projective plane, they can establish a direct key. Otherwise, there is a certain chance that they share one or more keys because the two key pools have some overlap. In order to establish direct keys, nodes can exchange the key IDs of the key rings, or they can broadcast a list of challenges as mentioned in [Eschenauer and Gligor (2002)] in order to secure the communication.
3.5.2.3 Path key establishment

If two nodes have key rings from different finite projective planes, they have to go through path key establishment phase to set up an indirect key. They need to find a number of intermediate nodes to establish a secure communication path from the source to the destination. Then the source node can send a secret securely to the destination and they will use this secret as their communication key. In this scheme, we only consider paths of 2 links (only one intermediate node) in order to reduce the communication overhead. As we will see from the analysis, using only 2-hop paths we can get a very high key connectivity that is close to 100%. 
CHAPTER 4. Performance evaluation

In this section, we give performance analysis for the two new schemes we proposed based on five performance metrics. We also include the analytical results for the three existing schemes. Besides analyzing the schemes individually, we also provide the performance comparison of the five schemes.

4.1 Performance metrics

To analyze and compare the performance of the key management schemes, we adopt the following criteria.

Key connectivity Key connectivity is the probability that two neighboring nodes can establish a communication key. We consider 1-hop and 2-hop connectivity in our analysis. 1-hop connectivity is defined as the probability that two neighboring nodes share a common key after the direct key establishment phase. 2-hop connectivity refers to the probability that two neighboring nodes share a key after the path key establishment phase. We assume that only 2-hop paths are established and used to send the communication key.

Scalability Usually when the network size grows, with the same key ring size, the 1-hop key connectivity decreases. Therefore it is essential to support large network while still hold the key connectivity requirements. One way is to increase the key ring size so that each node store more keys. However, since sensors are limited in memory resources, there is a tradeoff between the scalability and the storage complexity. This
problem can be considered from two aspects. For a certain scheme, we would like to know the maximum supported network size given a fixed key ring size and the minimum key ring size given a fixed network size.

**Resilience against node capture** The resilience against node capture is an essential performance metric for key pre-distribution schemes. Since most sensor networks are deployed in hostile environments in an unattended fashion, the sensor nodes can be easily captured physically by the adversary. Thus all the keys stored in the captured nodes are revealed. Since some of the remaining links may use these keys to communicate, they are compromised as well. If the scheme is not resilient, even if only few nodes are captured, a large proportion of the network is compromised. We evaluate the resilience against node capture by estimating the fraction of remaining links that are also compromised when $x$ nodes are captured.

**Communication overhead** Because of the limited bandwidth and power of sensor nodes, low communication overhead is desired for the key pre-distribution schemes. We evaluate communication overhead by the number and size of messages sent and received by a sensor node during both direct and path key establishment phases.

**Computational complexity** Since the sensor nodes have very low computation capability, low computational complexity is required for the key pre-distribution schemes. Computational complexity is analyzed in terms of the number of unit functions executed. Unit functions are functions such as Search, Hash, MAC, VecMul(size), PolyEval(count), etc [Camtepe and Yener (2005)].

### 4.2 Analytical result

#### 4.2.1 Key connectivity

In this section, we analyze the key connectivity of the five schemes we discussed in the previous section. Two types of connectivity are considered: 1-hop connectivity $C_1$
and 2-hop connectivity $C_2$, which are defined as follows:

- 1-hop connectivity $C_1$ - the probability that two neighboring nodes are able to find a common key after the direct key establishment phase.

- 2-hop connectivity $C_2$ - the probability that two neighboring nodes are able to find a common key after the path key establishment phase. We define the probability as the 2-hop connectivity since we only consider 2-hop paths in the path key establishment phase.

4.2.1.1 Random scheme

1-hop connectivity: 1-hop connectivity is given in [Eschenauer and Gligor (2002)]. It can be calculated as:

$$C_1 = 1 - \frac{((P-K)^2)}{(P-2K)!P!}.$$  

2-hop connectivity: If two neighboring nodes $u$ and $v$ do not share a key after the direct key establishment phase, they need to find a common neighbor $w$ to establish a path key, where $u$ and $w$ must share a key and so as $v$ and $w$.

The 2-hop connectivity is given in [Chan et al. (2003)]:

$$C_2 = C_1 + (1 - C_1)(1 - (1 - C_2)^0.5865n'),$$

where $n'$ is the expected number of neighbors of each node and $c = 0.5865n'$ is the number of nodes that are in both $u$ and $v$'s communication ranges [Chan et al. (2003)].

4.2.1.2 FPP-based symmetric scheme

Using the FPP-based symmetric scheme, every pair of nodes have exact one key in common. So every pair of nodes can communicate with each other after the direct key establishment phase. Therefore we have $C_1 = C_2 = 1$. 

4.2.1.3 FPP-complementary scheme

1-hop connectivity: A detailed analysis on the 1-hop connectivity is given in [Camtepe and Yener (2004)].

2-hop connectivity: The 2-hop connectivity $C_2$ can be calculated using the 1-hop connectivity $C_1$ similar to that for the random scheme: $C_2 = C_1 + (1 - C_1)(1 - (1 - C_1^{0.5865n'})$.

4.2.1.4 FPP-random scheme

1-hop connectivity: There are three types of neighboring node pairs $(u, v)$:

- $BB(u, v)$: Both nodes' key rings are from the core design.

- $BR(u, v)$: One node's key ring is from the core design and the other's is from random part.

- $RR(u, v)$: Both nodes' key rings are from random part.

If we use $C(u, v)$ to denote that $u$ and $v$ share at least one key, the 1-hop key probability $C_1$ is: $C_1 = P(C) = P(C|BB)P(BB) + P(C|BR)P(BR) + P(C|RR)P(RR)$.

Since the core design can support $N' = n^2 + n + 1$ out of $N$ nodes in the network and the random part supports $N - N'$ nodes, $P(BB)$, $P(BR)$ and $P(RR)$ can be calculated as follows.

$$P(BB) = \frac{N'(N' - 1)}{N(N - 1)},$$
$$P(BR) = \frac{2N'(N - N')}{N(N - 1)},$$
$$P(RR) = \frac{(N - N')(N - N' - 1)}{N(N - 1)}.$$

Since the core design uses finite projective plane, the probability that the two nodes sharing a common key is 1, therefore $P(C|BB) = 1$. As we have discussed for the
random scheme, \( P(C|RR) = 1 - \frac{(P-K)!}{(P-2K)!P!} \) [Eschenauer and Gligor (2002)]. So we only need to calculate \( P(C|BR) \). For two neighboring nodes \( u \) and \( v \), assume that \( u \)'s key ring is from the core design and \( v \)'s key ring is from the random part. The probability that \( v \) does not share any key with \( u \) is \( \frac{(P-K)!}{(P-2K)!P!} \). Thus \( P(C|BR) = 1 - \frac{(P-K)!}{(P-2K)!P!} \).

Therefore, \( C_1 = P(C|BB)P(BB) + P(C|BR)P(BR) + P(C|RR)P(RR) = \frac{N(N-1)}{N(N-1)} + (1 - \frac{(P-K)!}{(P-2K)!P!}) \times \frac{(N-N')(N+N'-1)}{N(N-1)}. \)

**2-hop connectivity:** As we have discussed in the random scheme, the 2-hop connectivity can be calculated based on the 1-hop connectivity as follows: \( C_2 = C_1 + (1 - C_1)(1 - (1 - C_1)^{0.5865n'}) \).

### 4.2.1.5 FPP-based two-party scheme

**1-hop connectivity:** We use a different method to calculate the 1-hop connectivity for FPP-based two-party scheme. The probability that two neighboring nodes share a common key is equal to the probability that two arbitrary nodes share a key since all the nodes are deployed randomly. Therefore, the probability can be calculated as the fraction of node pairs that share a common key. Since there are \( N = 2N' \) nodes in the network, there are \( \binom{N}{2} = \frac{N(N-1)}{2} = N'(2N' - 1) \) pairs of nodes. For any two nodes \( u \) and \( v \), if their key rings are from the same key pool, they always share exactly one key due to the construction of the finite projective plane. The number of node pairs is \( 2\binom{N'}{2} = N'(N' - 1) \) in this case. We now consider the case that \( u \) and \( v \) have key rings from different key pools. Since \( \rho N' \) keys are shared by the two key pools (overlapping keys) and every node carries \( K \) keys, the expected number of overlapping keys in each node is \( \frac{\rho N'}{N'} K = \rho K \). We first consider a node \( u \) whose key ring is from key pool \( P(K_1) \). \( u \) has \( \rho K \) overlapping keys. Since all the overlapping keys are at the same position in both key pools, there must exists another node \( v \), whose key ring is from \( P(K_2) \), carries the same shared keys as \( u \). Due to the construction of the finite projective plane, for each overlapping key in \( v \), exactly \( n = K - 1 \) nodes from \( P(K_2) \) also carry this key. Thus
\(\rho Kn\) nodes from \(P(K_2)\) carry one common overlapping key with node \(v\). Therefore, \(\rho Kn + 1\) nodes in \(P(K_2)\), including \(v\), share common keys with \(u\) from \(P(K_1)\). Since \(N'\) nodes have key rings from \(P(K_1)\), the expected number of node pairs \((u, v)\) with \(u\)'s key ring from \(P(K_1)\), \(v\)'s key ring from \(P(K_2)\), and \(u, v\) share a common key is \((\rho Kn + 1)N'\). Therefore, the total expected number of node pairs that share common keys can be calculated as \(N'(N' - 1) + (\rho Kn + 1)N'\). So the probability of two nodes share a common key is \(C_1 = \frac{N'(N' - 1) + (\rho Kn + 1)N'}{N'(2N' - 1)}\), with \(N' = n^2 + n + 1\).

2-hop connectivity: We use \(L(u, v)\) to denote that \(u\) and \(v\) can find a bridge \(w\) to set up a 2-hop secure path. Therefore the 2-hop connectivity \(C_2 = P(C) + P(L \land \neg C) = P(C) + P(L \land \neg C)(1 - P(C))\).

So we need to calculate \(P(L \land \neg C)\), which is the probability that \(u\) and \(v\) can set up a 2-hop secure path given that they do not share a common key. Consider a common neighbor \(w\) of nodes \(u\) and \(v\). \(w\)'s key ring must from the same FPP as \(u\) or \(v\). WLOG, we assume \(w\) and \(u\) are from the same FPP \(P(K_1)\). Thus \(w\) and \(v\) are from different FPPs. From the analysis for the 1-hop connectivity, we know that the probability that \(w\) and \(v\) share a key is given by \(\frac{\rho Kn + 1}{N'}\), which is also the probability that \(u\) and \(v\) can use \(w\) as an intermediate node to securely establish a key.

We use \(c\) to denote the number of common neighbors of \(u\) and \(v\). \(u\) and \(v\) can establish a 2-hop secure path if and only if at least one of these \(c\) nodes share common keys with both \(u\) and \(v\). Thus, the probability that \(u\) and \(v\) can establish a 2-hop secure path if they do not share any common key is given by: \(P(L \land \neg C) = 1 - (1 - \frac{\rho Kn + 1}{N'})^c\).

Since the expected number of common neighbors \(c\) is \(0.5865n'\), where \(n'\) is the expected number of neighbors of one node [Chan et al. (2003)], we have \(P(L \land \neg C) = 1 - (1 - \frac{\rho Kn + 1}{N'})^c = 1 - (1 - \frac{\rho Kn + 1}{N'})^{0.5865n'}\). And the 2-hop connectivity is \(C_2 = C_1 + (1 - C_1)(1 - (1 - \frac{\rho Kn + 1}{N'})^{0.5865n'})\).
4.2.2 Scalability

4.2.2.1 Random scheme

According to the description of the random scheme, an arbitrary key ring size $K$ can support sensor networks of any size. However, to maintain the global connectivity $P_c$ of the network, there are some restrictions on the relationship of network size $N$, key pool size $P$, and key ring size $K$ [Eschenauer and Gligor (2002)]. Global connectivity $P_c$ is defined as the probability that the whole network is connected [Eschenauer and Gligor (2002)]. For example, $P_c = 0.999$ means that 99.9% of the nodes share common keys with some of their neighbors.

From the analysis in [Eschenauer and Gligor (2002)], given the network size $N$, the expected number of neighbors of each node $n'$, and the desired $P_c$, we can find the required 1-hop probability $p$, which is the required 1-hop probability for the whole network to be connected with the probability $P_c$.

On the other hand, according to the previous discussion on 1-hop connectivity, the actual probability that two neighboring nodes share a key can be calculated as $p = 1 - \frac{(P-K)^2}{(P^2-K^2)}$. Therefore given the key pool size $P$, we can calculate the key ring size $K$ that is required to get the required 1-hop connectivity $p$. Similarly, given the key ring size $K$, we also can get the key pool size $P$. Moreover, given the key ring size $K$ and the key pool size $P$, we can get the 1-hop connectivity $p$. If we know the expected number of neighbors and the desired global connectivity $P_c$, we are able to get the maximum supported network size $N$.

4.2.2.2 FPP-based symmetric scheme

For a given sensor network size $N$, we can find the smallest prime power $n$ such that $n^2 + n + 1 \geq N$. To use the FPP-based symmetric scheme, we require every node to support a key ring size $K \geq n + 1$. In opposite, if the key ring size of each node is given
as $K$, we can find out the maximum supported network size using this scheme. We first find out the maximum prime power $n$ such that $n+1 \leq K$ and the maximum supported network size is $N = n^2 + n + 1$.

### 4.2.2.3 FPP-complementary scheme

The scalability of this scheme is discussed in [Camtepe and Yener (2004)]. Given the key ring size $K = n + 1$, the FPP-complementary design can support network sizes up to $\binom{n^2+n+1}{n+1}$.

### 4.2.2.4 FPP-random hybrid scheme

Using the FPP-random hybrid scheme, except the nodes that are supported by the core design, all the other nodes can randomly pick $K$ keys to form their key rings. Therefore, given a key ring size $K$, the FPP-random hybrid scheme is able to support networks of any size. However, it has the similar problem as the random scheme. Since part of the network is supported by the random scheme, the 1-hop key connectivity must be high enough to achieve desired global network connectivity $P_c$. Using the formula given in [Eschenauer and Gligor (2002)], we can calculate the desired 1-hop key connectivity $C_{ld}$ given the network size $N$, the expected number of neighbors $n'$, and the desired global network connectivity $P_c$ [Eschenauer and Gligor (2002)]. Meanwhile, following the analysis in the key connectivity section, we know that the real 1-hop connectivity is $C_{lr} = \frac{N'(N'-1)}{N(N-1)} + \left(1 - \frac{(P-K)!}{(P-2K)!P!}\right) \times \frac{(N-N')(N+N'-1)}{N(N-1)}$. Therefore we require $C_{lr} \geq C_{ld}$ to achieve the desired global network connectivity. Since $N' = n^2 + n + 1$ and $K = n + 1$, given $n$ and $P$, we can calculate the maximum supported network size $N$ by solving the inequality $C_{lr} \geq C_{ld}$. For this scheme, $n$ is restricted to only prime powers. Also, given $P$ and $N$, we are able to get $n$, and hence the key ring size $K$, based on the same inequality.
4.2.2.5 FPP-based two-party scheme

Let \( N \) be the network size. Using the FPP-based two-party scheme, we need to construct two finite projective planes \( P(K_1) \) and \( P(K_2) \). Each FPP includes \( N' \) points, where \( N' = n^2 + n + 1 \approx \frac{N}{2} \) and \( n \) is a prime power. Each block in the FPP includes \( K = n + 1 \) points. Therefore each node in the network needs to carry at least \( K = n + 1 \approx \sqrt{2N-3-1} \) keys.

Similarly, given a key ring of size \( K \), we can calculate the maximum supported network size \( N \). First we find out the largest prime power \( n \leq K - 1 \) and use \( n \) as the FPP parameter. Thus the number of points in one FPP will be \( N' = n^2 + n + 1 \) and the total number of sensor nodes that the scheme can support is \( N = 2N' \).

4.2.3 Resilience against node capture

The network resilience against node capture is defined as the fraction of links in the remaining network that are compromised if \( x \) nodes are captured. A remaining link is compromised if its communication key is in the key ring of a captured node. In real scenarios, links are related to the communication range and only exist between neighboring nodes. However, the communication range does not affect the resilience against node capture since we assume that all the nodes are randomly deployed and have same communication range. Therefore, in our analysis, we assume that the communication range is large enough and every common key of each pair of sensor nodes is a link. We first consider the case that only one node \( c \) is captured. For each key \( k \) in \( c \)'s key ring, we calculate the expected number of node pairs in the remaining network that have the common key \( k \). Thus if we know the expected total number of remaining links in the network, we are able to calculate \( P_{\text{one}} \), the expected resilience against node capture when one node is captured. Now if \( x \) nodes are captured, the expected resilience against node
capture can be estimated as \( P_r \approx 1 - (1 - P_{one})^x \). In this section, detailed analysis for \( P_{one} \) is given for every scheme.

### 4.2.3.1 Random scheme

Using the random scheme, every node randomly picks \( K \) keys from the key pool as its key ring. For every key \( k \) in the key pool, the expected number of nodes that have \( k \) in their key rings are \( \frac{KN}{P} \). Therefore, the number of node pairs that share key \( k \) is \( n_k = \binom{KN}{2} \). Since there are \( P \) keys in the key pool, the expected total number of links is \( Pn_k \). If one node \( c \) is captured, \( K \) keys are compromised and the expected number of links that are compromised is \( Kn_k \). The node \( c \) itself is involved in \( K(\frac{KN}{P} - 1) \) links. Therefore, the resilience against node capture when \( c \) is compromised can be calculated as 

\[
P_{one} = \frac{Kn_k - K(K(\frac{KN}{P} - 1))}{Pn_k - K(K(\frac{KN}{P} - 1))} = \frac{KN - 2P}{PN - 2P}.
\]

### 4.2.3.2 FPP-based symmetric scheme

The FPP-based symmetric scheme guarantees that every two nodes share exact one key. Therefore, the total number of links is \( \frac{N(N-1)}{2} \), \( N = n^2 + n + 1 \). Besides, using the FPP-based symmetric scheme, every key is in exact \( K = n + 1 \) key rings. Thus for every key \( k \) of the captured node \( c \), \( \frac{K(K-1)}{2} \) links use \( k \) as their communication key. Among them, \( K(K - 1) \) links involves node \( c \). Therefore, 

\[
P_{one} = \frac{K\frac{K(K-1)}{2} - K(K-1)}{\frac{N(N-1)}{2} - K(K-1)} = \frac{K-2}{N-2} = \frac{n-1}{n^2 + n - 1}.
\]

### 4.2.3.3 FPP-complementary scheme

Using FPP-complementary scheme, \( N' \) nodes use blocks from the core design as their key rings and the other \( N - N' \) nodes randomly pick their key rings from the \( k \)-subsets of complementary blocks. We use 'class 1' and 'class 2' to denote these two types of nodes. For every key \( k \) in the key pool, \( K = n + 1 \) class 1 nodes have \( k \) in their key rings. And \( N' - K = n^2 \) complementary blocks also have \( k \). Since the \( N - N' \) class 2 nodes
randomly pick $K$-subsets of the complementary blocks as their key rings, the probability that $k$ is in the key ring of some class 2 node is $\binom{n^2-1}{n^2+n+1} \times \frac{n^2}{n^2+n+1} = \frac{n^2+1}{n^2+n+1}$. Therefore, the expected number of class 2 nodes that have $k$ is given by $\frac{n^2+1}{n^2+n+1} (N-N') = \frac{K(N-N')}{N'}$ and the expected total number of nodes that have $k$ is $\frac{K(N-N')}{N'} + K = \frac{KN}{N'}$. Therefore, the expected number of links using key $k$ is $n_k = \binom{KN}{2}$. Since there are $N'$ keys in the key pool, the expected total number of links is $N'n_k$. For the captured node $c$, the expected number of links that use keys from $c$'s key ring is $Kn_k$. And $K(K + \frac{K(N-N')}{N'}) - 1)$ links involve node $c$. Thus the resilience against node capture when $c$ is captured is $P_{\text{one}} = \frac{Kn_k - K(K + \frac{K(N-N')}{N'}) - 1}{N'n_k - K(K + \frac{K(N-N')}{N'}) - 1} = \frac{KN-2N'}{N'N-2N'}$.

### 4.2.3.4 FPP-random hybrid scheme

For the FPP-random hybrid scheme, $N'$ keys are randomly selected from the key pool to construct the core design. Due to the construction, these keys are associated with more links than the remaining $P - N'$ keys. There are also two classes of nodes using this scheme. We also use 'class 1' and 'class 2' to distinguish between them. The key rings of class 1 nodes are from the core design while the key rings of class 2 nodes are randomly picked from the whole key pool. For any key belonging to the core design, the expected number of nodes that have this key is $K + \frac{K(N-N')}{P}$, and the expected number of links that associate with this key is $n_{k_1} = \binom{K+\frac{K(N-N')}{2}}{2}$. For any key that is not in the core design, the expected number of nodes that have the key is $\frac{K(N-N')}{P}$, and the expected number of links that use this key is $n_{k_2} = \binom{\frac{K(N-N')}{2}}{2}$. Therefore, the expected total number of links is $N'n_{k_1} + (P-N')n_{k_2}$.

If one node $c$ is captured, it can either be a class 1 node or a class 2 node. If node $c$ is a class 1 node, all its $K$ keys belong to the core design. Therefore, the expected number of links that use keys from $c$'s key ring is $Kn_{k_1}$. Among them, $K(K + \frac{K(N-N')}{P}) - 1)$ links involve the node $c$. Therefore, the fraction of compromised links in the remaining
network when \( c \) is captured can be estimated as\
\[
P_{one_1} = \frac{K_{n_k_1} - K(K + \frac{K(N-N')}{P} - 1)}{N'n_{k_1} + (P-N')n_{k_2} - K(K + \frac{K(N-N')}{P} - 1)}.
\]

If node \( c \) is from class 2, \( \frac{N''K}{P} \) of its keys are from core design and the expected number of links whose communication keys are from \( c \)'s key ring is \( \frac{K}{P}(N'n_{k_1} + (P-N')n_{k_2}) \). And among them, \( \frac{N''}{P}K(K + \frac{K(N-N')}{P} - 1) + \frac{P-N'}{P}K(\frac{K(N-N')}{P} - 1) = \frac{K(KN-P)}{P} \) links involve \( c \). Therefore, the fraction of compromised links in the remaining network when \( c \) is captured can be estimated as\
\[
P_{one_2} = \frac{K_{n_k_1} - K(K + \frac{K(N-N')}{P} - 1) - \frac{KKN-P}{P}}{N'n_{k_1} + (P-N')n_{k_2} - K(K + \frac{K(N-N')}{P} - 1)}.
\]

Since there are \( N' \) class 1 nodes and \( N-N' \) class 2 nodes, the resilience against node capture when one node is captured is\
\[
P_{one} = \frac{N'}{N}P_{one_1} + \frac{N-N'}{N}P_{one_2}.
\]

### 4.2.3.5 FPP-based two-party scheme

Since two key pools \( P(K_1) \) and \( P(K_2) \) are used for the FPP-based two-party scheme and they may overlap, the overlapping keys are associated with more links than the keys that belong to only one key pool. For any link in the network, if the two nodes of the link have key rings from different key pools, it must use an overlapping key. As we have analyzed in the key connectivity section, the expected number of such links in the network is \( (\rho Kn + 1)N' \). If the two nodes are from different key pools, there must be a link between them and the total number of such links is \( N'(N'-1) \). Therefore, the total expected number of links in the network is \( (\rho Kn + 1)N' + N'(N'-1) \).

Now we only need to calculate the number of links that use overlapping keys with both nodes' key rings from the same key pool. As we have discussed, the expected number of overlapping keys in each node is \( \rho K \). For each overlapping key in node \( u \), there are \( n = K - 1 \) nodes that share this key with \( u \) and their key rings are from the same key pool as node \( u \). Since there are \( N' \) keys in a key pool, for the \( N' \) nodes that have key rings from the same key pool, \( \frac{\rho KnN'}{2} \) links use an overlapping key as their common key. Therefore, the total expected number of links that share overlapping keys is given by \( (\rho Kn + 1)N' + \frac{\rho KnN'}{2} \times 2 = (2\rho Kn + 1)N' \). And the expected number of
links that share non-overlapping keys is \( N'(N' - \rho Kn - 1) \).

If a node \( c \) is captured, among its \( K \) keys, the expected number of overlapping keys are \( \rho K \). Therefore, the number of links that associate with keys in \( c \)'s key ring can be calculated as 
\[
\frac{\rho K}{pN'} (2\rho Kn + 1) N' + \frac{K - \rho K}{2N' - pN'} N'(N' - \rho Kn - 1) = \frac{K(3\rho Kn - \rho^2 Kn + (1 - \rho)N' + 1)}{2 - \rho}.
\]
Among these links, \( \rho K (2K - 1) + (1 - \rho) K(K - 1) = Kn + \rho K^2 \) links involve \( c \). Therefore, the resilience against node capture when \( c \) is compromised is 
\[
P_{\text{once}} = \frac{K(3\rho Kn - \rho^2 Kn + (1 - \rho)N' + 1)(Kn + \rho K^2)}{2 - \rho}.
\]

4.2.4 Communication overhead

Since algorithms of direct and path key establishment phases are similar for all the schemes, here we give a brief analysis for all of them. During the direct key establishment phase, each innocent node needs to broadcast 1 message to its \( n' \) neighbors. Each message includes the key IDs of all the keys in the node's key ring. Suppose one key ID has \( d \) bits, the message length is \( dK \) bits since there are \( K \) keys in each node's key ring. Each node also receives \( n' \) messages from its neighbors, each contains \( dK \) bits.

During the path key establishment phase, each innocent node unicasts 1 message to the neighbors with whom it shares a direct key. The expected number of messages sent is \( n'C_1 \), where \( n' \) is the expected number of neighbors and \( C_1 \) is 1-hop connectivity. The message includes the key ID of both source and destination nodes. Meanwhile, each node receives \( n'C_1 \) messages from its connected neighbors. After checking its own key ring, a node replies with a message to each request indicating whether it shares a common key with the destination node. So there are \( 2n'C_1 \) messages sent and \( 2n'C_1 \) messages received in total. The messages are very short. Therefore the communication overhead is dependent on the 1-hop connectivity. If 1-hop connectivity is high, less communication is required in the path key establishment phase. For instance, for the FPP-based symmetric scheme, there is no message sent or received in the third phase,
since every pair of nodes can establish a direct key.

4.2.5 Computational complexity

For all these schemes, sensor nodes do not need to perform much computation. All the computation in the key pre-distribution phase is done offline. Most computational complexity is generated by functions like 'search' in both direct and path key establishment phases. Every time a node needs to check whether it has a common key with some other node, it has to perform a 'search' in its own key ring. For all the five schemes, during the direct key establishment phase, each node must perform $n'$ 'searches' for its $n'$ neighbors. However, in the path key establishment phase, a node performs a 'search' only when it is chosen by another two nodes to establish a path key between them. Also, in this phase, all the messages sent and received are encrypted and decrypted, which also contribute to the computational complexity. Therefore, less communication in the path key establishment phase means less computation is required for the sensor nodes. In other words, the computational complexity is also dependent on 1-hop connectivity. An effective method to reduce the computational complexity is to increase 1-hop connectivity.

4.3 Performance analysis

4.3.1 Comparison of schemes

Since applications vary from one to another, there is no one-size-fits-all key pre-distribution scheme for sensor networks. For instance, some applications have more concerns on the resilience performance while some others may focus on the scalability problem. By comparing the schemes, we are able to see the advantages and disadvantages of every scheme, which allows us to choose tradeoffs and to apply the most suitable scheme to different applications. In this section, we compare the above five key pre-
distribution schemes on the performance of key connectivity, resilience against node
capture, and scalability.

4.3.1.1 Key connectivity

Key connectivity is considered as the probability that two neighboring nodes are able
to establish a pair-wise key after deployment. Good key pre-distribution schemes must
have high 1-hop key connectivity to save communication in the path key establishment
phase. Besides, 2-hop key connectivity needs to be close to 1 since it is desired that
every pair of neighboring nodes can be connected by at most 2-hop after the path key
establishment phases.

The FPP-based symmetric scheme guarantees that two neighboring nodes can estab-
lish a communication key after the direct key establishment phase. The communication
and computation overhead in path key establishment phase is avoided using this scheme.
Therefore, FPP-based symmetric scheme has the best key connectivity among the five
schemes. For the rest four schemes, we calculate both the 1-hop and 2-hop connectivity
based on the same network size and key ring size. Based on the same network size
$N = 10,000$ and key ring size $K = 72$, the analytical results of 1-hop and 2-hop con-
nectivity for the four schemes are given in the 4.1. Note that for FPP-complementary
scheme, the key pool size is $P = n^2 + n + 1 = 5113$; for FPP-based two-party scheme,
if the overlapping factor $\rho = 0.2$, the key pool size is $P = (2 - 0.2)(n^2 + n + 1) = 9203$.
If $\rho = 1$, $P = 5113$. In this case, the two finite projective planes are identical and the
1-hop connectivity is 1. For random and FPP-random scheme, the key pool size can be
of any size.

From table 4.1 we are able to conclude that the random scheme has the poorest
1-hop connectivity among the five schemes with the same network size, key ring size
and key pool size. Compared to the random scheme and the FPP-random schemes, the
FPP-based two-party scheme has the best 1-hop connectivity. As the overlapping factor
Scheme | 1-hop connectivity | 2-hop connectivity
---|---|---
Random \((P = 9203)\) | 0.4332 | 0.9854
Random \((P = 5113)\) | 0.6424 | 0.9999
FPP-based symmetric \((P = 5113)\) | 1 | 1
FPP-random \((P = 9203)\) | 0.5846 | 0.9997
FPP-random \((P = 5113)\) | 0.7396 | 0.9999
FPP-complementary \((P = 5113)\) | 0.7523-0.9883 | 0.9999
FPP-based two-party \((P = 9203)\) | 0.6 | 0.9921
FPP-based two-party \((P = 5113)\) | 1 | 1

Table 4.1 Key connectivity \((N = 10,000, K = 72)\)

As \(p\) increases, the number of overlapping keys also increases and the 1-hop connectivity becomes higher. However, when there are more overlapping keys, the network becomes more vulnerable since the resilience against node capture becomes worse. So we need to choose the overlapping factor carefully based on different applications.

### 4.3.1.2 Resilience against node capture

For the security concern, we prefer the resilience against node capture, which is the fraction of remaining links compromised, to be as low as possible when some nodes are captured. It is obvious that the fraction decreases when the key pool size \(P\) increases and key ring size \(K\) decreases. However, there are tradeoffs between resilience and key connectivity since key connectivity is lower with larger key pool and/or smaller key ring. Consider the example we have used to compare the key connectivity. The network size is \(N = 10,000\) and key ring size is \(K = 72\). The resilience against node capture is given in table 4.2.

From table 4.2, we can see that with the same key ring size and key pool size, the random scheme achieves the best resilience against node capture, followed by the FPP-
based two-party scheme and the FPP-random scheme. The FPP-complementary scheme has a worse resilience than the FPP-random scheme since its key pool size is fixed with certain key ring size while the FPP-random scheme is able to use larger key pools and hence to achieve better resilience. With a key ring size of 72, the FPP-based symmetric scheme can only support a network of size 5113 and the resilience against node capture is close to that of the FPP-complementary scheme. To support a network of size 10,000 using the FPP-based symmetric scheme, some nodes must have the same key ring, which will make the resilience against node capture much worse.

Although with the same key ring size and key pool size, the random scheme outperforms the FPP-based two-party and the FPP-random scheme on resilience against node capture, it is not the same case when we take key connectivity into consideration. Table 4.3 demonstrates the resilience against node capture for 10,000 nodes with same key ring size and expected 1-hop key connectivity but different key pool sizes for the five schemes.

For the FPP-based symmetric and the FPP-complementary scheme, once the key ring size and network size are fixed, the expected 1-hop key connectivity is also fixed. Therefore, we cannot compare the resilience of them with the other three schemes directly. For the remaining three schemes, with the same expected 1-hop key connectivity,
the FPP-based two-party scheme achieves the best resilience against node capture, followed by the FPP-random scheme, while the random scheme has the worst resilience.

4.3.1.3 Scalability

Scalability of different schemes is compared based on the maximum supported network size using the same key ring size. From the discussion in section 4.2.2, with a fixed key ring size, both the random scheme and the FPP-random scheme can support a network of any size. However, to support same number of sensor nodes and achieve same 1-hop key connectivity, the FPP-random scheme uses a larger key pool than the random scheme, which implies better resilience against node capture. For example, with key ring size 72, to support a network of 10,000 nodes and to achieve a 1-hop key connectivity of 0.6, the resilience against node capture is 0.71 using the FPP-random scheme, which is better than 0.67 when using the random scheme. For the other three schemes, the maximum network sizes are related to key ring size. From our discussion in section 4.2.2, the FPP-complementary scheme has better scalability than the FPP-based two-party scheme and the FPP-based symmetric scheme has the worst scalability.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>50 captured</th>
<th>200 captured</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random ((P = 5730))</td>
<td>0.4633</td>
<td>0.917</td>
</tr>
<tr>
<td>*FPP-based Symmetric ((P = 5113, N = 5113, C_1 = 1))</td>
<td>0.4982</td>
<td>0.9366</td>
</tr>
<tr>
<td>FPP-random ((P = 8644))</td>
<td>0.4097</td>
<td>0.8786</td>
</tr>
<tr>
<td>*FPP-complementary ((P = 5113, C_1: 0.7523-0.9883))</td>
<td>0.5030</td>
<td>0.9390</td>
</tr>
<tr>
<td>FPP-based two-party ((P = 9203))</td>
<td>0.3552</td>
<td>0.8272</td>
</tr>
</tbody>
</table>

Table 4.3 Resilience against node capture \((N = 10,000, K = 72, C_1 = 0.6)\)
4.3.2 Experimental results

4.3.2.1 Experimental settings:

To evaluate the schemes we have discussed and to verify our analytical result, we use the following configurations for experiments. We assume that there are 10,000 sensors in total in the deployment area, each with communication radius of 10m. The expected number of each sensor node is 30. We designed three experiments to test the affect of different parameters on the key connectivity and resilience against node capture of our proposed schemes.

- **Experiment I**: In this experiment, we focus on the 1-hop key connectivity when key pool size varies. We fix the network size to be 10,000 nodes and key ring size to be 72. Since key pool size is related to the overlapping factor $\rho$ in our FPP-based two-party scheme, we use the change in overlapping factor to represent the change in key pool size. In this experiment, we run our simulations when the overlapping factor varies from 0.2 to 1. We test three different schemes using various key pool size: the random scheme, the FPP-based random scheme, and the FPP-based two-party scheme, because for the FPP-based symmetric and the FPP-complementary schemes, key pool size is fixed if key ring size is fixed.

- **Experiment II**: The second experiment is about the resilience against node capture when key pool size varies. Total number of nodes is set to be 10,000, key ring size is also fixed to be 72 and the number of node captured is set to be 50.

- **Experiment III**: We test the resilience against node capture with various number of captured nodes. The experiment is also performed on 10,000 nodes, with key ring size 72 and key pool size 9203 (overlapping factor $\rho = 0.2$). The number of node captured varies from 50 to 200.
Besides, two more experiments are performed to evaluate the performance of our proposed schemes with different network sizes.

- **Experiment IV**: We change the network size from 2,000 to 10,000 nodes in this experiment. Therefore, the corresponding minimum key ring size changes from 38 to 72. This experiment evaluates the 1-hop connectivity in networks of different sizes. For all three schemes, we use the key pool of the FPP-based two-party scheme in this experiment. Therefore, the change of key pool size with different network size is related to the key ring size. We use overlapping factor $\rho = 0.4$ in this experiment.

- **Experiment V**: Although for a fixed network size, we are able to find a minimum key ring size, we are also interested in the performance of our schemes when larger key ring sizes are applied. In other words, we want to test whether larger key ring size can improve the 1-hop connectivity and resilience against node capture. Therefore, in this experiment, we use a key ring size of 72 (which is the minimum key ring size for 10,000 nodes in our two-party scheme) with different network sizes from 1,000 to 10,000 to evaluate our schemes. The key pool size is 9203 and other parameters are the same as described in previous experiments.

**4.3.2.2 Experiment results:**

As we have discussed, the key pool size $P$ has significant effect on the performance of key connectivity and resilience against node capture. Figure 4.1 shows the 1-hop key connectivity with different key pool sizes ($P = (2 - \rho)(n^2 + n + 1)$ and $\rho = 0.1, 0.2, \cdots, 1$) for the random scheme, the FPP-random scheme, and the FPP-based two-party scheme. The FPP-based symmetric scheme and the FPP-complementary scheme are not included because these two schemes require fixed key pool size for the above settings.
Note that the 1-hop connectivity increases as the key pool size decreases (overlapping factor $\rho$ increases). Compared to the random scheme, the two FPP-based schemes have improved the 1-hop key connectivity greatly because they use the symmetric design which guarantees some overlap among the key rings of sensor nodes. The FPP-based two-party scheme is even better since all the key rings are under the symmetric design while for the FPP-random scheme, some of the key rings are still randomly selected. For the extreme case, where $\rho = 1$, the FPP-based two-party scheme consists of two identical finite projective planes. In this case, the two-party scheme can be compared to the FPP-based complementary scheme since they have the same key pool size. The two-party scheme achieves 100% 1-hop connectivity in this case, which is definitely higher than the FPP-complementary scheme. Besides, the two-party scheme is more flexible than the FPP-based symmetric scheme or the FPP-complementary scheme, since we can adjust the overlapping factor $\rho$ to meet the performance requirements of different applications.

Figure 4.2 shows the resilience against node capture for different key pool sizes in the same setting that we have described. We assume 50 nodes are captured.

Figure 4.3 compares the resilience of node capture of different schemes with different number of nodes being captured. We assume that the key pool size is $P = 9203$ (for the two-party scheme, $\rho = 0.2$).

As showed in the graph, resilience of the FPP-random scheme is a little worse than that of the FPP-based two-party scheme. This is because that, for the FPP-random scheme, some keys are more likely to be in the key rings since the core design only uses part of the whole key pool. Also, the resilience of the random scheme is a little better than that of the FPP-based two-party scheme as showed in the graph. However, in this case, the 1-hop key connectivity of the random scheme is much worse than that of the FPP-based two-party scheme. As we have discussed in 4.3.1.2, with the same desired 1-hop key connectivity, the FPP-based two-party scheme has better resilience against
Figure 4.1 Key connectivity

Figure 4.2 Resilience against node capture when size of key pools changes
node capture than both the FPP-random scheme and the random scheme. Therefore, it is obvious that the FPP-based two-party scheme performs better than the random scheme. Compared to the FPP-random scheme, the FPP-based two-party scheme also has better resilience and better key connectivity. Therefore comes the conclusion that the FPP-based two party scheme has the best performance among the three schemes.

Figure 4.4 shows the result of experiment IV. For our the FPP-based two-party scheme, as the network size increases to 10,000, the minimum key ring size reaches 72, which is less than two times the key ring size when the network size is 2,000. Meanwhile, the 1-hop connectivity maintains the same. Therefore, our two-party scheme is scalable.

Figure 4.5 represents the influence of different network sizes on the key connectivity of the three schemes we have discussed. With the same key ring size $K = 72$ and same key pool size $P = 9203$, the 1-hop key connectivity of the FPP-based two-party scheme is almost the same for networks of different sizes. However, for the FPP-random scheme, as illustrated in the graph, it achieves 100% key connectivity with small networks and the connectivity reduces as the number of nodes increases. This is because when the network is small, most nodes pick their key rings from the base design which guarantees 100% 1-hop key connectivity. However, the resilience of node capture is higher for networks with less nodes, as represented in figure 4.6.

The resilience against node capture of the three schemes is given in figure 4.6. As the number of nodes increases, the resilience of the random scheme becomes worse while the resilience of the FPP-random scheme improves a little. However, the FPP-based two-party scheme maintains the same resilience against node capture for different network sizes.

As we can see from the result, for certain network size, larger key ring size does not bring much benefit for the performance of the FPP-based two-party scheme. However, larger key ring size improves the 1-hop connectivity of the FPP-random scheme while at the same time decreases the resilience against node capture. Therefore, for the FPP-
Figure 4.3  Resilience against node capture with the changes of the number of nodes captured

Figure 4.4  Key connectivity with the change of the network size and key ring size.
Figure 4.5  Key connectivity with the change of the network size

Figure 4.6  Resilience against node capture with the change of the network size
based two-party scheme, we should always use the minimum key ring size for a certain network size. For the FPP-random scheme, we may decide what key ring size to use based on different applications.
CHAPTER 5. Conclusion and future work

In this paper, we have discussed five different key pre-distribution schemes for wireless sensor network, which fall into three categories: probabilistic scheme, deterministic scheme, and hybrid scheme. The first three schemes are proposed by different researchers in the past several years, including the random scheme [Eschenauer and Gligor (2002)], the FPP-based symmetric scheme [Camtepe and Yener (2004); Lee and Stinson (2005)], and the FPP-complementary scheme [Camtepe and Yener (2004)]. The random scheme is a probabilistic approach. It is simple and easy to implement while suffers from the low key connectivity problem. To improve the key connectivity, researches proposed the deterministic FPP-based symmetric scheme using finite projective plane. This scheme achieves perfect key connectivity (100%). But it has some problems with scalability and resilience against node capture. A hybrid FPP-complementary scheme was proposed in [Camtepe and Yener (2004)] to improve the performance of the FPP-based symmetric scheme. However, the resilience against node capture in this scheme is not very satisfactory. Therefore, we have proposed two novel key pre-distribution schemes in this paper. Our goal is to solve the scalability problem of the FPP-based symmetric scheme while improve the key connectivity and maintain a good resilience against node capture.

The FPP-random scheme is a combination of the symmetric scheme and the random scheme. The core design of the FPP-random scheme is a finite projective plane that can support some of the nodes in the sensor network. Other nodes will pick their key rings using the method in the random scheme. With a given key ring size, the FPP-random scheme can support a network of any size. Therefore, it successfully solve the
scalability problem of the FPP-based symmetric scheme. Besides, since the key pool size in this scheme is adjustable, it is more flexible and achieves better resilience against node capture than the FPP-based complementary scheme. However, when the network size is large and the key ring size is small, this scheme performs more like the random scheme and the key connectivity is not very good. The FPP-based two-party scheme uses two finite projective planes instead only one as in the FPP-based symmetric scheme. Therefore, compared to the FPP-based symmetric scheme, our two-party scheme can support more sensor nodes and achieves better scalability. Although the key connectivity of our scheme is not as good as the FPP-based symmetric scheme, it outperforms the other four schemes. Especially when the overlapping factor is 1, the two-party scheme achieves 100% key connectivity. Besides, the resilience against node capture of the two-party scheme is also better than the FPP-based symmetric, FPP-complementary, and FPP-random schemes.

Future work can be pursued to further improve the FPP-based two-party scheme. For example, more finite projective planes can be added to achieve better scalability and resilience against node capture. However, add more finite projective planes will decrease the key connectivity of the network. More detailed analysis is required to find an optimized solution. Besides, for all the five schemes, resilience against node capture is not very good when large number of nodes are captured. More future research need to be performed to improve the resilience against node capture in these key pre-distribution schemes.


ACKNOWLEDGEMENTS

This thesis would not have been feasible without the support of many people.

First and foremost, I would like to thank my major advisor, Dr. Johnny S. Wong, whose academic guidance and continuing support helped me throughout the entire time of research and the writing of this thesis. I would also like to thank my co-advisor, Dr. Wensheng Zhang, who read numerous revisions and offered valuable suggestions. Thanks must also go to Dr. Richard Ng and Dr. Ling Long from Mathematics Department, Iowa State University, who have helped me a lot in formalizing the solution and theoretical analysis in this thesis. I would also like to thank my committee members for their efforts and contributions to this work: Dr. Wallapak Tavanapong and Dr. Richard Ng.