A KEY STORAGE AND PATH KEY EFFICIENT DIAGONAL-BASED GROUPING FOR WIRELESS SENSOR NETWORK

MD ASIF KHAN
Bachelor of Science, North South University, 2011

A Thesis
Submitted to the School of Graduate Studies
of the University of Lethbridge
in Partial Fulfillment of the
Requirements for the Degree

MASTER OF SCIENCE

Department of Mathematics and Computer Science
University of Lethbridge
LETHBRIDGE, ALBERTA, CANADA

A KEY STORAGE AND PATH KEY EFFICIENT DIAGONAL-BASED GROUPING FOR WIRELESS SENSOR NETWORK

MD ASIF KHAN

Date of Defense: April 22, 2016

Dr. Hua Li Supervisor	Associate Professor	Ph.D.
Dr. Jacqueline E. Rice Committee Member	Associate Dean	Ph.D.
Dr. Gongbing Shan Committee Member	Professor	Ph.D.
Dr. Howard Cheng Chair, Thesis Examination Committee	Associate Professor	Ph.D.

Dedication

To my family and friends

Abstract

Research into the security of wireless sensor networks, often referred to as WSN, has always been a great challenge due to the limited resources and a rich domain of active research. Recently developed probabilistic key predistribution for WSN groupings are not entirely secure. If an adversary can compromise a certain number of sensors, s/he could reconstruct the keys for rest of the sensors. The objective of this thesis was to develop a storage-efficient and low pathkey consuming grouping scheme for a wireless sensor network. In this thesis, a diagonal-based grouping is proposed to improve the security and performance of key distribution based on the work conducted by Liu, Ning, and Du [1]. Two different types of grouping schemes are presented: diagonal-based grouping and diagonal$_{\min}$ grouping. The step-by-step implementation of these groupings in several types of network orientations is also described. This thesis examines the proposed grouping schemes in terms of the key storage and the length of the pathkey. Finally, the outcomes of this thesis demonstrate that the proposed grouping is more key-storage efficient than are the existing schemes. If there is a lot of data flow across the diagonals, the proposed grouping would demonstrate efficient key utilization.

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor, Dr. Hua Li, for his continuous guidance, motivation, and immense knowledge throughout the journey of my MSc program. His direction, valuable opinions, and efforts have led me along the path of my thesis. I could not imagine having a better advisor and mentor for my research.

In addition to my advisor, I would like to thank Dr. Jacqueline E. Rice for her encouragement, insightful comments, and difficult questions, which certainly helped me to ascertain my research objective. She took the time to review my thesis and her valuable feedback has helped me to widen my research from various perspectives. My sincere thanks also to Dr. Gongbing Shan for his thoughtful advice and insightful feedback regarding my thesis work.

My appreciation also extends to my fellow graduate students, family members and people at the Teaching Centre whose cooperation has made this path easier for me. Last but not the least, I would like to thank my wife for supporting me spiritually throughout writing this thesis and in my life in general.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Research into the security of wireless sensor networks, often referred to as WSN, has always been a great challenge due to the limited resources and a rich domain of active research on account of sensors' reduced cost and the ease of simple deployment in remote and hard-to-reach areas without communication infrastructures [3].

WSN helps people to collect and analyze the desired data automatically and remotely in different environments. For example, sensors can monitor environmental factors [4], military purposes [5], and healthcare [6], as well as being used in industrial applications such as instrumentation and predictive maintenance, which involves tracking the state of the machine, noise levels, lighting conditions, mechanical stress levels of attached objects, and other properties [7, 8]. These features of WSN have attracted many researchers to work on various issues related to these types of networks. In order to protect the transmitted data and authenticate the wireless sensor node, various security protocols and schemes have been proposed [9, 10, 11, 12]. On the other hand, routing strategies and wireless sensor network modelings are also receiving much attention [13, 14, 15, 16]. However, the security issues associated with WSN have yet to receive extensive focus. In particular, key distribution schemes (KDS) play a significant role in security in sensor networks. Various security services, such as authentication and encryption, rely on KDS. However, due to the inherited resource constraints of the sensor nodes, key management for WSN is a significant problem [17].

1.1 Motivation for the Thesis

Various types of groupings of WSN are available. The orientation of these groupings ranges from circular to hexagonal, but none of the work that we are aware of has tried to exploit the concept of the diagonal. Therefore, I have proposed and developed a key storage and pathkey efficient diagonal-based grouping for wireless sensor networks. I have also addressed two key problems during my research. They are:

1. Dynamic key distribution: Many studies have focused on the distribution of keys among sensors. However, very few researchers have addressed the limitation of how the keys will be distributed after calculation.

2. Neighbour discovery: Neighbour discovery is a common issue when dealing with the grouping of the sensor network. I propose two methods to identify any sensor neighbours after deployment.

Researchers have used the path length (number of hops to reach from one sensor to the destination) as a property for performance evaluation [18]. I have refined this concept by considering not only the number of hops, but also the number of keys used to encrypt data in these hops.

1.2 Main Contribution

The main contributions of this thesis are as follows:

1. **New grouping scheme:** I introduce a new type of grouping scheme for WSN, based on the concept of the diagonal, which requires significantly fewer keys to be stored and a shorter path across the diagonals. I propose two types of models based on this concept:

   (a) Diagonal based grouping

   (b) $\text{Diagonal}_{\text{min}}$ grouping

2. **Considering dynamic key distribution:** Liu et al. [1] did not consider dynamic distribution of pairwise keys. I discuss two possible options to address this issue.

3. **Efficient neighbour discovery:** Two methods have been presented for neighbour discovery. One of these requires no communication overhead while the other requires establishing a secure communication.

4. **Efficient key utilization:** During performance analysis I considered the path in terms of required the number of keys. This approach allows a better understanding of network performance. My proposed grouping utilizes a moderate amount of keys for encryption with fewer keys in the storage.

5. **Network based performance evaluation:** The proposed diagonal based grouping was implemented in different network sizes. The performance of these different sized networks is presented here.

1.3 Thesis Organization

The remainder of the thesis is organized as follows. In Chapter 2, a general overview of WSN is presented, which is required to understand the remainder of this thesis. This chapter focuses on the application and inherited limitations of WSN such as deployment, maintenance, and vulnerabilities. It also summarizes the key distribution requirements for a secure WSN, which is followed by an explanation of symmetric key cryptography and hash functions.

Chapter 3 reviews the literature on various types of key distribution schemes. The proposed diagonal-based grouping scheme is discussed in Chapter 4, including various algorithms and mathematical models. This chapter also addresses the issues with dynamic key distribution and neighbour discovery. The performance analysis and evaluation of the proposed scheme are shown in Chapter 5. This chapter also offers a comparison of our results with previous results. I evaluate the groupings in terms of key storage, number of

hops and key usage for encryption. Finally, the thesis is concluded and future plans are proposed in Chapter 6.

Chapter 2

Background Study

The term Wireless Sensor Network (WSN) is usually used to describe a network that consists of hundreds to thousands of tiny sensors with wireless communication capability. They can often communicate over a short distance [9]. A typical sensor consists of:

- a processing unit,
- a storage unit,
- a sensing unit,
- a power unit, and
- a wireless transceiver.

Although sensors are generally designed for specific tasks, a WSN may monitor multiple parameters such as temperature, light, sound, or acceleration by combining different types of sensors.

A standard architecture for a WSN [19] is shown in Figure 2.1, which consists of various types of sensors including a humidity sensor and a temperature sensor to sense multiple environmental parameters, as well as a base station. An individual sensor exchanges data with neighbouring sensors within its communication range and relays these data to the base station. A base station can be considered to be an access point or a gateway to another data processing or management unit. [19].

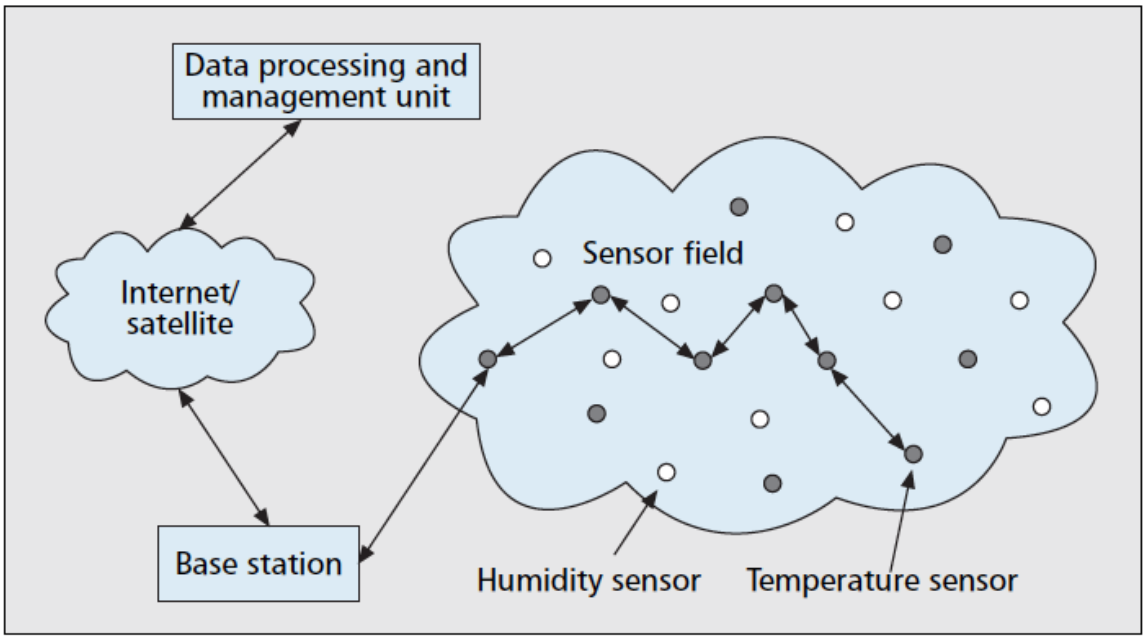


Figure 2.1: Simplified architecture of a wireless sensor network [19].

2.1 Limitations of Wireless Sensor Networks

The following features of WSN make the grouping scheme design complicated and extremely challenging. They are discussed below.

The inefficiency of public key cryptosystems: The use of public key algorithms, such as Diffie-Hellman key agreement [20] or RSA signatures [21] is often inefficient due to the sensors having inadequate computation and power resources. Currently, these operations may require a lot of time for sensor nodes to perform [22, 23]. This delay exhibits a vulnerability to denial of service (DoS) attacks [24]. The simplest DoS attack sends extra unnecessary packets to exhaust the resources available to the victim node. This attack prevents genuine sensors from accessing authorized services or resources of the WSN.

Bootstrapping: Bootstrapping is important because it is the phase in which the sensors in a network are made aware of the presence of all or some of the other entities in

the network. Bootstrapping is challenging because an efficient synchronization is required to avoid data collisions.

Physical capture: In many applications, sensor nodes are deployed in public or hostile conditions in large numbers. Thus, it is not cost effective for manufacturers to make each sensor node tamper resistant. An adversary may take advantage of this vulnerability, and might compromise the cryptographic keys undetectably.

Knowledge about deployment: It is difficult for the sensor network protocols to know ahead of time which nodes will be inside the communication range of each other after they are deployed via random scattering. Even if the nodes are deployed manually, it becomes costly, as a large number of nodes are involved to predetermine the location of every single node.

Limited memory resources: The capacity of key storage memory of a sensor is limited. It is impossible for a single node to establish a common key with all the other sensors of the same network.

Limited bandwidth and transmission power: Conventional sensor networks operate on minimal bandwidth [25]. The communication of large blocks of data is particularly expensive as transmission reliability is often low.

Reliance on base stations exposes vulnerabilities: The base station plays a vital role in a WSN. Base stations are often used as a reliable source of trust because they have high computational power. However, they are few and expensive. This draws the attention of attackers to the base station and minimizes the application of the security protocol.

2.1.1 Security Requirements for WSN

A security service is a process or communication service that is provided by a system to ensure a special set of protection to system resources. The service implements security policies that are carried out by security mechanisms [2]. Wireless networks are more

vulnerable to attacks than are wired ones because WSNs transmit data by broadcasting, have nodes with limited resources, and often operate in uncontrolled environments where nodes are left unattended. Security requirements for WSNs are similar to those for ad-hoc networks [26, 27]. Therefore, WSNs have the following general security requirements [28]:

Availability: Availability is the characteristic of a network or a network resource being accessible and able to be utilized upon demand by an authorized system entity [2].

Authentication: Authentication is concerned with ensuring that communication is trustworthy [2]. This means authenticating other nodes and base stations before the actual data exchange.

Integrity: Integrity is required to ensure that the message or the data under consideration are not modified.

Confidentiality: Confidentiality is required to provide privacy to the wireless communication channels to prevent eavesdropping or any passive attacks [2].

Non-reputation: Non-reputation is the process of identifying and preventing malicious nodes from hiding their activities.

Survivability: Survivability is the ability to provide a minimum level of service in the event of power loss, failures or attacks.

Degradation of security services: This is the ability to change security levels as resource availability changes.

In addition to these general requirements, WSNs have the following specific requirements [19]:

Resilience to capture: An adversary can physically attack a sensor node after deployment.

Resilience to node replication: An attacker may insert additional hostile nodes into a WSN, leading to a severe attack.

Node revocation or participation: The addition of new sensor network or removal of a misbehaving or suspicious sensors should be dynamic for any existing sensor network.

Scalability: As the number of sensors grows in, the security level decreases [2]. It is essential that a network is scalable.

Based on these security requirements, the ideal key distribution system should have the following desirable properties [29, 30]:

1. minimal memory cost, as this is directly proportional to the number of keys stored,
2. minimum path length between two nodes that are inside the communication range of each other,
3. high resilience to node compromise,
4. ability of two nodes within communication range to establish secure communication with high probability using a common key, and
5. low computation and communication overheads to establish any common key.

Various key management schemes have been proposed [21, 31, 32] to address these security issues and to fulfill the requirement to develop a strong system. They can be categorized according to three types [33]:

1. **Centralized key management**: In this approach the key management is carried out by a trusted central server. The most common example is Kerberos [31]. Every single sensor in the network only trusts itself and the trusted server, which is most commonly a basestation equipped with powerful hardware. We can take Security Protocols for Sensor Networks (SPINS) [32] as an example of a centralized key management system. In SPINS, any two nodes that wish to communicate with each other

must execute a basestation-intermediate-based protocol for authentication. Although the centralized nature makes key management simple and convenient, it forces nodes to communicate frequently with the basestation. This causes the batteries of the sensor nodes close to the base station to drain quickly. The base station is also a high priority target for attackers.

2. **Public key infrastructure**: As mentioned earlier in this chapter, well-established key cryptosystems including Diffie-Hellman key agreement [20] and RSA signatures [21] offer a high level of security. However, current asymmetric encryption and decryption methods require intensive computation. It is usually not practical to perform the operations of an asymmetric cryptography system in WSN because of the resource limitations of the sensor nodes.

3. **Key predistribution**: In this category keys are loaded into the sensor nodes before the actual deployment [34, 35]. This requires a key-management protocol to distribute keys into nodes to provide the communication nodes with a common session key. This category is suitable for the limited hardware of sensor nodes [36]. Consequently, the grouping scheme proposed in this thesis falls under this type of scheme.

2.2 Symmetric Cryptography

A symmetric encryption scheme has five components (Figure 2.2):

Plaintext: This is the original understandable message or data that are provided to the algorithm as input.

Encryption algorithm: The encryption algorithm performs different substitutions and transformations on the plaintext.

Secret key: The secret key is another input to the encryption algorithm. The key does not depend on the plaintext and the algorithm. The output of the algorithm will vary

depending on the particular key because the explicit substitutions and transformations executed by the algorithm depend on the key.

Ciphertext: This is the scrambled message generated as output depending on the plaintext and the secret key. For any given message, different keys will result in two different ciphertexts. A ciphertext should appear to be a random stream of meaningless data.

Decryption algorithm: This algorithm is the reverse form of the encryption algorithm. It combines the ciphertext and the secret key and generates the original plaintext.

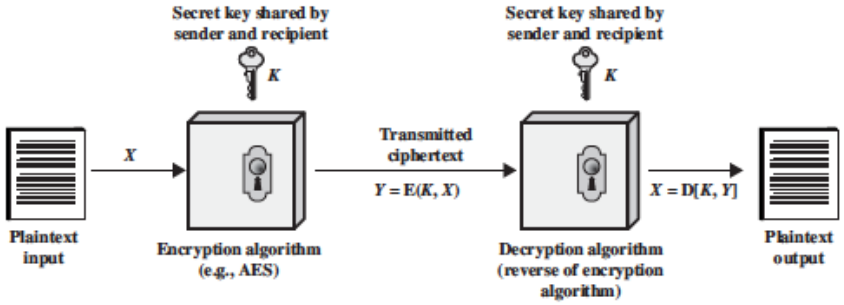


Figure 2.2: A simplified model of conventional encryption [2].

2.2.1 Cryptographic Hash Functions

A hash function H takes a variable-length block of data M as input and outputs a fixed-size hash value h:

$$h = H(M)$$

A *good* hash function will result in evenly distributed and apparently random output after applying a large set of inputs. In other words, the principal objective of a hash function is data integrity. If there is any alteration to any bit or bits in M, it is highly likely that the alteration will result in a shift of the hash code with high probability. The type of hash function required for security applications is known as a cryptographic hash function. [2]

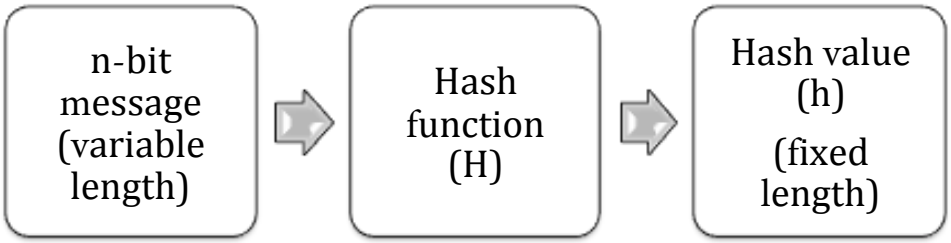


Figure 2.3: Basic steps of a hash function [2].

A cryptographic hash function is an algorithm with extremely efficient one-way and collision-free properties. It is computationally impossible to find either a data object that maps to a pre-defined hash output or two data objects that map to the identical hash output. Because of these features, hash functions are often used to discover whether or not data has changed [2].

2.2.2 Secure Hash Algorithm (SHA)

The *secure hash algorithm* (SHA) is the most extensively used hash function [2]. It was developed by the National Institute of Standards and Technology (NIST). In 1993 it was published as a federal information-processing standard. After finding weaknesses in SHA (SHA-0), a revised version was developed as FIPS 180-1 in 1995 and was referred to as SHA-1[2]. SHA-1 generates a hash value of 160 bits. NIST later produced an updated version of the standard, FIPS 180-2, which outlined three new versions of SHA. Their hash value lengths are 256, 384, and 512 bits respectively, known as SHA-256, SHA-384, and SHA-512. These hash algorithms are collectively known as SHA-2. They use a similar underlying structure, as well as the same kinds of modular arithmetic and logical binary operations as SHA-1 (Table: 2.1).

Table 2.1: Parameters of SHA-1 [2].

	SHA -1
Message digest size	160
Message size	$< 2^{64}$
Block size	512
Word size	32
Number of steps	80
Security	80

Note: All sizes are measured in bits

Chapter 3

Literature Review

There are three main phases or steps involved in the key distribution of WSN. Before the actual deployment, sensors are preloaded with keys, which is called the *key predistribution* phase. The next phase is the *key establishment* phase in which the deployed sensors try to discover whether they share a common key with their neighbours. If a common key is absent, then a path key is established using an intermediate node, which relays data between the two nodes trying to communicate. This phase is called the *path key establishment* phase.

Various schemes are used for key predistribution. One scheme is to load all the nodes with one master key, which will ensure optimal storage. However, one compromised node will make the entire network insecure. Furthurmore, as all the nodes have the same key, there is no way of authenticating individual nodes or revoking selected keys upon detection of sensor capture. Alternatively, each pair of nodes could share a unique key, also known as the pairwise key. This scheme results in an entirely secure network, as one compromised key will only reveal its counterpart and not the compromised keys of any other nodes. However, this scheme requires high storage. That is, for a network with n nodes, each node has to store $n-1$ pairwise key and the total number of keys in the system is $\frac{n!}{2!(n-2)!} = \binom{n}{2}$ which makes it impractical for a large sensor network. Storing many keys does not solve the actual demand, as nodes will communicate with pre-specified nodes covering a small node neighbourhood. Adding and removing, as well as re-keying sensors, is expensive and complicated, and every operation will require a broadcast message.

A probabilistic key distribution scheme was proposed by Eschenauer and Gligor [9]

whereby two nodes possess one or more (m) keys with specific probabilities that are drawn at random from a large key pool (K). The value of m and the size of K can be selected to ensure any pair of nodes has a certain probability of sharing a key.

A variation of this scheme was proposed by Chan, Perrig and Song [10] and exploits the idea that not all nodes may be within the communication range of each other, and only nodes in close proximity will communicate with each other sharing, at least, q predistributed keys to calculate the pairwise key. This threshold-based idea causes each node to store $k < (N-1)$ pairwise keys. However, every node needs to store not only pairwise keys but also all the node identifiers, which presents a significant storage cost of $O(klogN)$. This random pairwise key scheme addresses the storage problem yet provides excellent key resilience. In an extended version of this scheme proposed by Chan et al. [10], two sensor nodes are required to share minimum q predistributed keys to calculate a pairwise key. Chan et al. [11] also developed a random pairwise key scheme for key establishment called PIKE. PIKE utilizes peer sensor nodes as entrusted agents. Its most valuable feature is that compromised nodes in this scheme do not lead to the compromise of any key shared directly between two non-compromised sensor nodes.

The closest (location-based) pairwise key predistribution scheme [12] is an alternative to the random pairwise key scheme [10]. This scheme improves key connectivity by taking advantage of location information. Some schemes have taken advantage of various grouping schemes [37, 38, 39] for wireless sensor networks, notably multivariate polynomial-based grouping [40] and cluster-head-based grouping [41].

Numerous techniques that take advantage of the sensor's location information to improve key pre-distribution have been introduced recently [42, 43]. Two similar threshold-based schemes were developed independently by Liu, Ning, Du, and their colleagues [12, 44, 45, 46, 47]. Some of these schemes made the assumption that the sensor nodes' locations can be predetermined to a certain extent [48] while others assumed that the locations of the sensor nodes could be determined after deployment [49].

Sensor nodes are deployed after predistributing the keys. The deployment can either be arranged or random [9], based on the application; for example, as square grids [11], triangular or hexagonal grids [48], or in groups [44, 45]. However, there is still much concern regarding which key management is better when comparing probabilistic or deterministic [50] deployment.

This thesis describes a diagonal-based grouping to improve security and performance of key predistribution based on the work done by Liu, Ning, and Du [1]. Sensor nodes in the same group generally reside close to each other after deployment, as discovered by Liu et al. In [1] Liu et al. did not assume any prior knowledge of any deployment points, as was the case in some previous research [46, 51] in which deployment points were predetermined. In practice, many factors will affect the final deployment position of a sensor node. However, Liu et al. [1] have argued that when sensors [9] are usually deployed in groups together at the same time, it is very likely that they will be affected in a similar manner by the same set of factors. For instance, when several sensor nodes are dropped from an airplane to a remote location, all of them will be affected similarly by the location and the velocity of the plane. This results in an increase of the probability of the sensors in the same group being close to each other.

Chapter 4

Proposed Diagonal-Based Grouping Using Hash Key

I propose an efficient scheme to establish row groups (R) and diagonal groups (D) for a pairwise key establishment. The notations used in this grouping are described in Table 4.1.

Table 4.1: Notation used in diagonal-based grouping.

Symbol	Description
n	Number of rows
m	Number of the sensor in each row
R_a [1]	a^{th} row group
D_a	a^{th} diagonal group
ID(x)	Current ID of sensor x
ID(y)	Current ID of sensor y
$E_{K_i}(M)$	Message M is encrypted using key K_i
hash	SHA-1 hash function

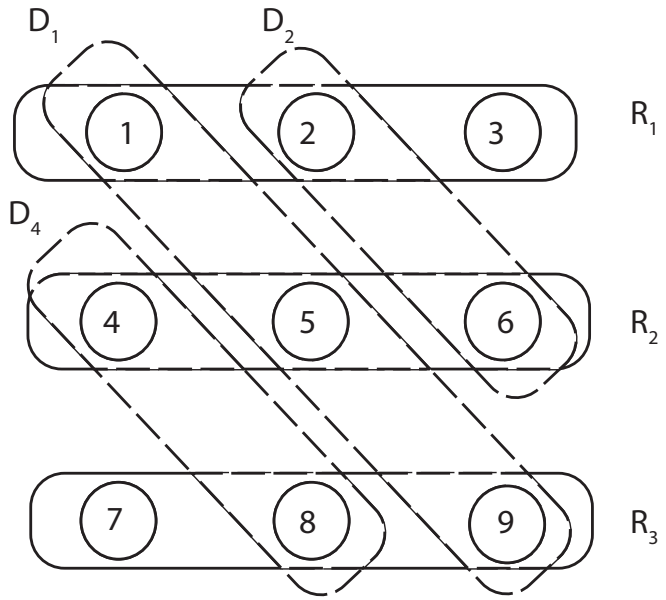


Figure 4.1: Diagonal-based grouping.

4.1 Predistribution

4.1.1 Group Construction

In the proposed grouping the first step is to divide an $n \times m$ network into different row groups $(R_1, R_2, \ldots, R_i)$ as shown in Figure 4.1 where n is the number of rows and m is the number of sensors in each row. Each row group (R_a) contains the sensors with the IDs

$$R_a = \Big\{ (a-1)m + b \Big\}_{\,1 \le a \le n\,,\, 1 \le b \le m} \tag{4.1}$$

where a and b are positive integers.

The network is also divided into diagonal groups $(D_1, D_2, \ldots, D_j)$ as shown in Figure 4.1. Diagonal groups contain sensors with the IDs

$$D_a = \Big\{ a + (b-1)(m+1) \Big\}_{\,1 \le a \le m(n-1)\,,\, 1 \le b \le m} \tag{4.2}$$

where a and b are positive integers.

The requirements on these diagonal groups are:

1. In any diagonal group the maximum ID number of a sensor is $ID_{\max} = n \times m$.

2. Each diagonal group includes exactly one sensor node from each row group and there are no common sensor nodes between any two different diagonal groups. Expressed mathematically, $R_a \cap D_a = 1$ and $D_a \cap D_b = \varnothing$.

3. All diagonal groups have size greater than one: $|D| > 1$.

4. Any pair of adjacent ID(p) and ID(q) in diagonal groups will always belong to adjacent row groups, $\{\text{ID}(p), \text{ID}(q)\} \in \{R_a, R_{a+1}\}$. For example, if the first ID of two adjacent IDs of a diagonal group is from R_1 then the second ID must belong to R_2.

4.1.2 Group Instantiation

The proposed grouping is applied in three different types of network orientations:

1. When the number of row groups (n) and the number of sensors in each row (m) is the same, $n = m$. For instance, $n = m = 3$ as shown in Figure 4.2 (a).

2. When the number of row groups (n) is smaller than the number of sensors in each row (m), $n < m$. For instance, $n = 3$ and $m = 4$ as shown in Figure 4.2 (b).

3. When the number of row groups (n) is greater than the number of sensors in each row (m), $n > m$. For instance, $n = 4$ and $m = 3$ as shown in Figure 4.2 (c).

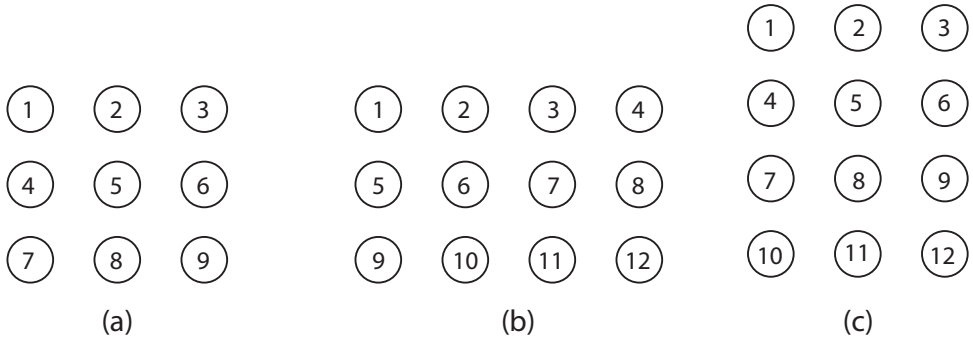


Figure 4.2: Examples of the three different types of network orientations.

Row group instantiation:

Using Equation 4.1, setting $a = 1,2,3$ and $b = 1,2,3$ for $n = m = 3$, the row groups will be as shown in Figure 4.3 (a).

$$R_1 = \{1,2,3\}$$

$$R_2 = \{4,5,6\}$$

$$R_3 = \{7,8,9\}$$

For $n = 3$, $m = 4$, setting $a = 1,2,3$ and $b = 1,2,3,4$, the row groups will be as shown in Figure 4.3(b).

$$R_1 = \{1,2,3,4\}$$

$$R_2 = \{5,6,7,8\}$$

$$R_3 = \{9,10,11,12\}$$

And for $n = 4$, $m = 3$, setting $a = 1,2,3,4$ and $b = 1,2,3$, the row groups will be as shown in Figure 4.3(c).

$$R_1 = \{1,2,3\}$$

$$R_2 = \{4,5,6\}$$

$$R_3 = \{7,8,9\}$$

$$R_4 = \{10,11,12\}$$

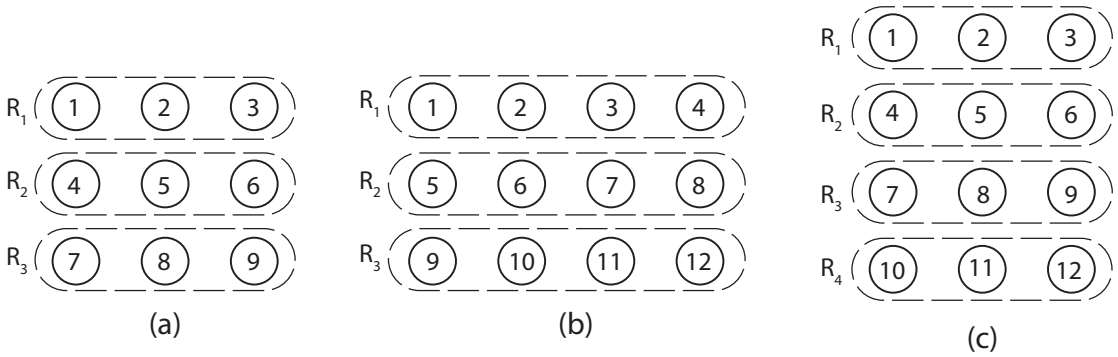


Figure 4.3: Row group instantiation for different network orientations.

Diagonal group instantiation:

Using Equation 4.2, setting the maximum value of $a = m(n-1) = 3 \times 2 = 6$ for $n = m = 3$, the diagonal groups will be as shown in Figure 4.4(a).

$$D_1 = \{1 + (j-1) \cdot 4\} = \{1, 5, 9\}$$

$$D_2 = \{2 + (j-1) \cdot 4\} = \{2, 6\}$$

$$D_3 = \{3 + (j-1) \cdot 4\} = \{3\}\ D_3 \text{ is discarded since } |D_3| < 2$$

$$D_4 = \{4 + (j-1) \cdot 4\} = \{4, 8\}$$

$$D_5 = \{5 + (j-1) \cdot 4\} = \{5, 9\}\ D_5 \text{ is discarded because } \{5, 9\} \in D_1$$

$$D_6 = \{6 + (j-1) \cdot 4\} = \{6\}\ D_6 \text{ is discarded since } |D_6| < 2$$

For $n = 3$ and $m = 4$, setting the maximum value of $a = m(n-1) = 4 \times 2 = 8$, the diagonal groups will be as shown in Figure 4.4(b).

$$D_1 = \{1 + (j-1) \cdot 5\} = \{1, 6, 11\}$$

$$D_2 = \{2 + (j-1) \cdot 5\} = \{2, 7, 12\}$$

$$D_3 = \{3 + (j-1) \cdot 5\} = \{3, 8\}$$

$$D_4 = \{4 + (j-1) \cdot 5\} = \{4\}\ D_4 \text{ is discarded since } |D_4| < 2$$

$$D_5 = \{5 + (j-1) \cdot 5\} = \{5, 10\}$$

$$D_6 = \{6 + (j-1) \cdot 5\} = \{6, 11\}\ D_5 \text{ is discarded because } \{6, 11\} \in D_1$$

$$D_7 = \{7 + (j-1) \cdot 5\} = \{7, 12\}\ D_5 \text{ is discarded because } \{7, 12\} \in D_2$$

$$D_8 = \{8 + (j-1)\cdot 5\} = \{8\}\ D_8 \text{ is discarded since } |D_8| < 2$$

And for n = 4 and m = 3, setting the maximum value of $a = m(n-1) = 3 \times 3 = 9$, the diagonal groups will be as shown in Figure 4.4(c).

$$D_1 = \{1 + (j-1)\cdot 4\} = \{1,5,9\}$$
$$D_2 = \{2 + (j-1)\cdot 4\} = \{2,6\}$$
$$D_3 = \{3 + (j-1)\cdot 4\} = \{3\}\ D_3 \text{ is discarded since } |D_3| < 2$$
$$D_4 = \{4 + (j-1)\cdot 4\} = \{4,8,12\}$$
$$D_5 = \{5 + (j-1)\cdot 4\} = \{5,9\}\ D_5 \text{ is discarded because } \{5,9\} \in D_1$$
$$D_6 = \{6 + (j-1)\cdot 4\} = \{6\}\ D_6 \text{ is discarded since } |D_6| < 2$$
$$D_7 = \{7 + (j-1)\cdot 4\} = \{7,11\}$$
$$D_8 = \{8 + (j-1)\cdot 4\} = \{8,12\}\ D_5 \text{ is discarded because } \{8,12\} \in D_4$$
$$D_9 = \{9 + (j-1)\cdot 4\} = \{9\}\ D_9 \text{ is discarded since } |D_9| < 2$$

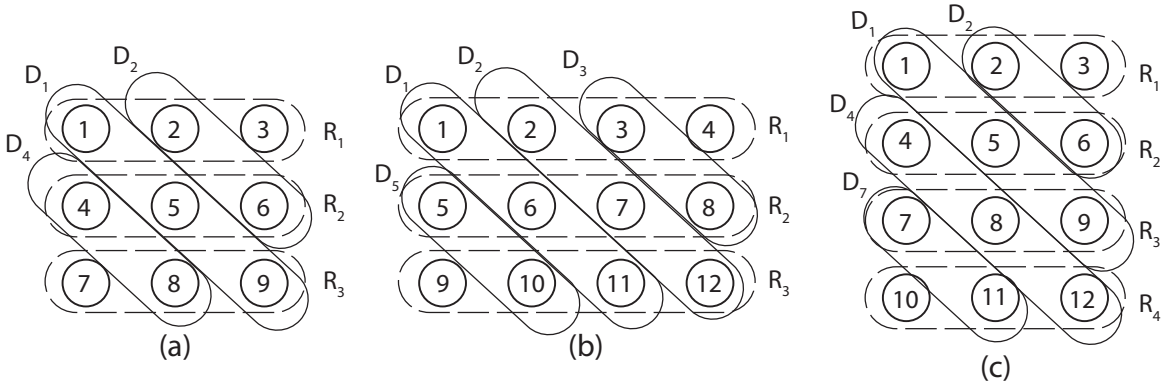


Figure 4.4: Diagonal group instantiation for different types of network orientations.

A second proposed grouping, diagonal$_{\text{min}}$ requires fewer keys than the diagonal grouping. The construction of diagonal$_{\text{min}}$ is the same as the diagonal grouping discussed earlier, except that for diagonal$_{\text{min}}$ we consider only the first and last row of the network. Thus the IDs of sensor nodes for each R_a are

$$R_a = \Big\{(a-1)m + b\Big\}_{\{b = 1,\ldots, m,\ a = 1,\ n\}} \tag{4.3}$$

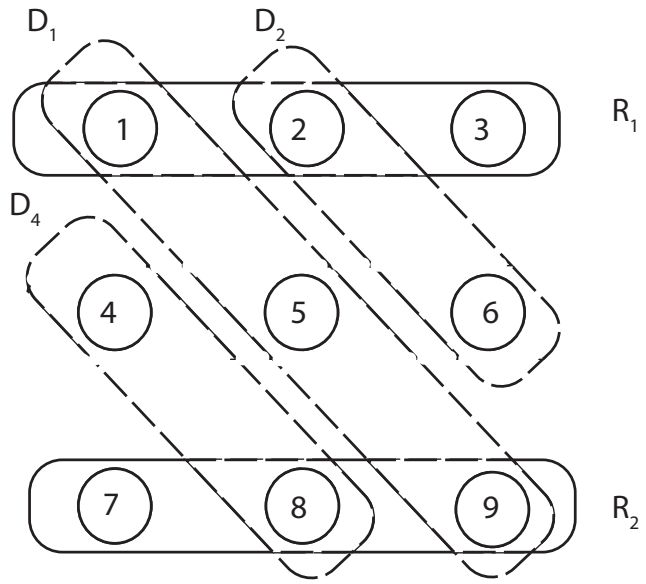


Figure 4.5: Diagonal$_{\text{min}}$ grouping.

From Figure 4.5, $R_1 = \{1, 2, 3\}$, and $R_3 = \{7, 8, 9\}$.

For the network where $n > m$ we must add an extra row to maintain network connectivity.

$$\begin{aligned} \text{if n is even, } R_a &= \Big\{(a-1)m+b\Big\}_{\{\text{b = 1,..., m, a = 1, } \frac{n}{2}\text{, n}\}} \\ \text{if n is odd, } R_a &= \Big\{(a-1)m+b\Big\}_{\{\text{b = 1,..., m, a = 1, } \frac{(n+1)}{2}\text{, n}\}} \end{aligned} \tag{4.4}$$

4.2 Storing Row and Diagonal Group Values

Each sensor saves the assigned row and diagonal value in a small array, which costs only few kilobytes of storage. The first element of the array corresponds to the row value and the second element of the array corresponds to the diagonal value. This small array will help to find common rows or diagonal groups between two sensors. For example, the

array values for the sensors illustrated in Figure 4.1 are as follows,

$$\text{Sensor 1: } Row = 1, Diagonal = 1 \rightarrow \{1,1\}$$

$$\text{Sensor 2: } Row = 1, Diagonal = 2 \rightarrow \{1,2\}$$

$$\text{Sensor 3: } Row = 1, Diagonal = 0 \rightarrow \{1,0\}$$

$$\text{Sensor 4: } Row = 2, Diagonal = 3 \rightarrow \{2,3\}$$

$$\text{Sensor 5: } Row = 2, Diagonal = 1 \rightarrow \{2,1\}$$

$$\text{Sensor 6: } Row = 2, Diagonal = 2 \rightarrow \{2,2\}$$

$$\text{Sensor 7: } Row = 3, Diagonal = 0 \rightarrow \{3,0\}$$

$$\text{Sensor 8: } Row = 3, Diagonal = 3 \rightarrow \{3,3\}$$

$$\text{Sensor 9: } Row = 3, Diagonal = 1 \rightarrow \{3,1\}$$

This array value will give enough information to identify the potential neighbours to establish secure communication. For instance, the group array of sensor 5 is $\{2,1\}$. This means sensor 5 belongs to row 2 and diagonal 1. If there is no group or row assigned to a sensor, then the value becomes zero. For example, sensor 7 belongs to row 3 but it does not belong to any diagonal. Thus, the group array value is $\{3, 0\}$.

4.3 Neighbour Discovery

It is important that all sensors know their neighbours. This could be achieved in several ways:

1. Immediately after deployment each sensor broadcasts its ID. Sensors within the communication range will listen and store this ID in their neighbour list.

2. Sensors can find out their neighbour(s) from the hash keys that they are distributed with (discussed later in this section).

3. Two sensors can compare their group arrays to find out whether they are neighbours

or not. Any two sensors are neighbours if the row values are the same and their IDs differ by one. Alternatively, if the diagonal values are the same and the row values differ by one then they are also neighbours. This can be expressed as follows:

For two sensors with IDs p, q and group arrays $[R_p, D_p]$, $[R_q, D_q]$ respectively, the requirements to be neighbours are either:

(a) $R_p = R_q$ and $|ID(p) - ID(q)| = 1$, or

(b) $D_p = D_q$ and $|R_p - R_q| = 1$.

4.4 Distribution of Hash Keys in Diagonals

For any two sensors x and y belonging to the same diagonal group, the algorithm to determine a shared key is as follows:

Data: IDs of sensor x and y

Result: Distribution of master and hash keys in diagonals

initialization;

while *(ID(x) - ID (y))mod (m+1) = 0* **do**

- **if** $ID(x) < ID(y)$ **then**
  - $i \leftarrow [\frac{(ID(y)-ID(x))}{(m+1)}] mod(2)$;
  - **if** $i \neq 0$ **then**
    - Store K_x in sensor x;
    - Store hash key $H(K_x||ID(y))$ in sensor y;
  - **else**
    - Store K_y in sensor y;
    - Store hash key $H(K_y||ID(x))$ in sensor x;
  - **end**
- **else**
  - $i \leftarrow [\frac{(ID(x)-ID(y))}{(m+1)}] mod(2)$;
  - **if** $i \neq 0$ **then**
    - Store K_y in sensor y;
    - Store hash key $H(K_y||ID(x))$ in sensor x;
  - **else**
    - Store K_x in sensor x;
    - Store hash key $H(K_x||ID(y))$ in sensor y;
  - **end**
- **end**

end

Algorithm 1: Distribution of hash keys in diagonals.

As an example let us apply Algorithm 1 to the network as shown in Figure 4.1:

(a) Node 1 and node 5 are in the same diagonal group and node 1 wants to communicate with node 5. Here $1 < 5$ and $(5-1)mod(3+1) = 4mod(4) = 0$ and $\frac{5-1}{3+1} = 1$ (not divisible by 2). In this case, node 1 already stored K_1 and node 5 stores the hash key $H(K_1||5)$.

(b) Node 1 and node 9 are in the same diagonal group. Here $1 < 9$ and $(9-1)mod(3+1) = 8mod(4) = 0$ and $\frac{9-1}{3+1} = 2$ (divisible by 2). In this case, node 9 already stored K_9 and node 1 stores the hash key $H(K_9||1)$.

(c) Node 2 and node 6 are in the same diagonal group. Here $2 < 6$ and $(6-2)mod(3+1) = 4mod(4) = 0$ and $\frac{6-2}{3+1} = 1$ (not divisible by 2). In this case, node 2 already stored K_2 and node 6 stores the hash key $H(K_2||6)$.

(d) Node 4 and node 8 are in the same diagonal group and node 8 wants to communicate with node 4. Here $8 > 4$ and $(5-1)mod(3+1) = 4mod(4) = 0$ and $\frac{5-1}{3+1} = 1$ (not divisible by 2). In this case, node 4 already stored K_4 and node 8 stores the hash key $H(K_4||8)$.

4.5 Distribution of Hash Keys in Rows

For two sensors x and y that belong to the same row group, the algorithm to find a shared key is as follows:

Data: IDs of sensor x and y

Result: Distribution of master and hash keys in rows

initialization;

while *(ID(y) - ID (x))* $< m$ **do**

  if $ID(x) < ID(y)$ **then**

    $i \leftarrow [ID(y) - ID(x)] mod(2)$;

    if $i \neq 0$ **then**

      Store K_x in sensor x;

      Store hash key $H(K_x||ID(y))$ in sensor y;

    else

      Store K_y in sensor y;

      Store hash key $H(K_y||ID(x))$ in sensor x;

    end

  else

    $i \leftarrow [ID(x) - ID(y)] mod(2)$;

    if $i \neq 0$ **then**

      Store K_y in sensor y;

      Store hash key $H(K_y||ID(x))$ in sensor x;

    else

      Store K_x in sensor x;

      Store hash key $H(K_x||ID(y))$ in sensor y;

    end

  end

end

Algorithm 2: Distribution of hash keys in rows.

Let us apply Algorithm 2 to the network as shown in Figure 4.1 as an example:

(a) Node 1 and node 2 are in the same row group and node 1 wants to communicate with node 2. Here $1 < 2$ and $(2-1) = 1$ which is not divisible by 2 and is less than m. Thus, node 1 already stores K_1 and node 2 stores the hash key $H(K_1||2)$.

(b) Node 1 and node 3 are also in the same row group and node 1 wants to communicate with node 3. Here $1 < 3$ and $(3-1) = 2$ which is divisible by 2 and is less than m. Thus, node 3 already stores K_3 and node 1 stores the hash key $H(K_3||1)$.

(c) Node 4 and node 6 are in the same row group and node 4 wants to communicate with node 6. Here $4 < 6$ and $(6-4) = 2$ which is divisible by 2 and is less than m. Thus, node 6 already stores K_6 and node 4 stores the hash key $H(K_6||4)$.

(d) Node 5 and node 4 are in the same row group and node 5 wants to communicate with node 4. Here $5 > 4$ and $(5-4) = 1$ which is not divisible by 2 and is less than m. Thus, node 4 already stores K_4 and node 5 stores the hash key $H(K_4||5)$.

(e) Node 9 and node 7 are in the same row group and node 9 wants to communicate with node 7. Here $9 > 7$ and $(9-7) = 2$ which is divisible by 2 and is less than m. Thus, node 9 already stores K_9 and node 7 stores the hash key $H(K_9||7)$.

(f) Node 9 and node 8 are in the same row group and node 9 wants to communicate with node 8. Here $9 > 8$ and $(9-8) = 1$ which is not divisible by 2 and is less than m. Thus, node 8 already stores K_8 and node 9 stores the hash key $H(K_8||9)$.

In applying the above rules to Figures 4.1 and 4.5 we obtain the following key storage shown in Tables 4.2, 4.3, and 4.4.

Figures 4.6 and 4.7 show the hash key distribution among sensors for diagonal-based and diagonal$_{\text{min}}$ grouping respectively. An arrow coming out of the sensor means its hash key is stored in another sensor and an arrow on the sensor means the sensor contains a hash key.

Table 4.2: Key allocation for diagonal-based grouping for a 3×3 network (total 23).

Node number	1	2	3	4	5	6	7	8	9
Master Key	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9
Hash Key	3,9	1	2	6	1, 4	2,5	9	4, 7	8,5
Total	3	2	2	2	3	3	2	3	3

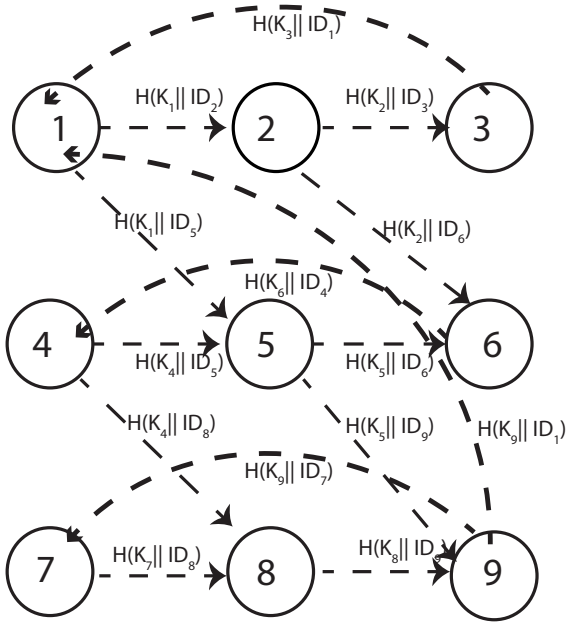


Figure 4.6: Distribution of hash keys in diagonal-based grouping for a 3×3 network

Table 4.3: Key allocation for diagonal$_{\text{min}}$ grouping for a 3×3 network (total 20).

Node number	1	2	3	4	5	6	7	8	9
Master Key	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9
Hash Key	3,9	1	2	0	1	2	9	4, 7	8,5
Total	3	2	2	1	2	2	2	3	3

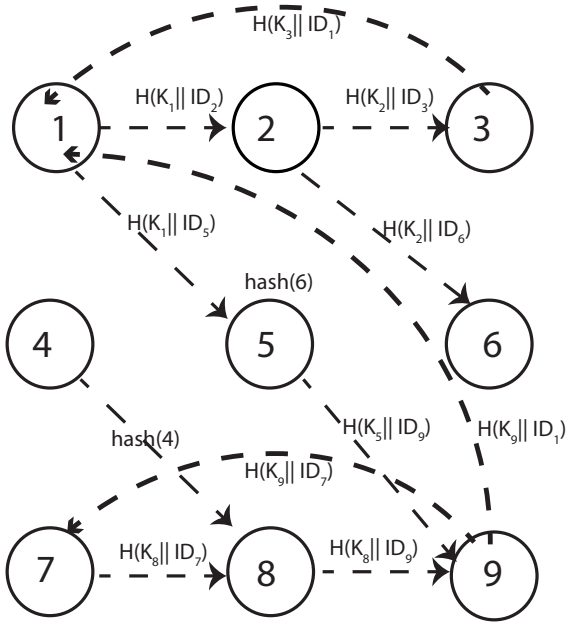


Figure 4.7: Distribution of hash keys in diagonal$_{\text{min}}$ grouping for a 3×3 network

Table 4.4: Key allocation for diagonal-based grouping for a 3×4 network (total 38).

Node number	1	2	3	4	5	6	7	8	9	10	11	12
Master Key	K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}	K_{11}	K_{12}
Hash Key	3,11	1, 4,12	2	1, 3	7	5, 8, 1	6, 2	5, 7, 3	11	9, 12, 5	10, 6	11, 7,9
Total	3	4	2	3	2	4	3	4	2	4	3	4

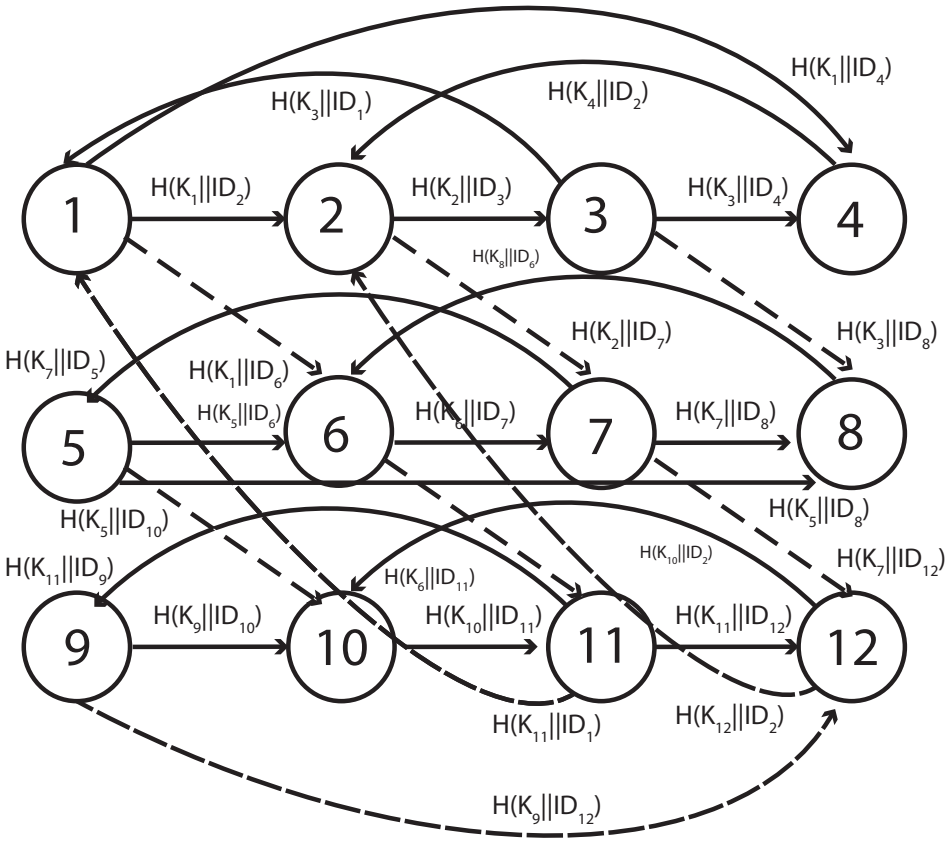


Figure 4.8: Distribution of hash keys in diagonal-based grouping for a 3×4 network.

4.6 Pairwise Key Distribution

Two pairwise key distribution methods can be applied to insert the pairwise hash keys into the sensor nodes.

1. The pairwise keys are generated in software and stored in the corresponding sensor nodes at the time of programming and configuration of the sensor nodes.

2. Since each sensor has been pre-loaded with a master key, we can use the master key to encrypt the pairwise key when transmitting the pairwise key from the base station. In this case, the pairwise key can also be changed in real time by changing the master key in the sensor node. For example, every few months the master keys are generally changed and the pairwise keys can be updated at the same time.

For example, we consider Figure 4.9, where the basestation (bs) is able to communicate with sensors 1, 4, and sensor 7 only. In order to insert the pairwise keys into sensor 3 let M be an encrypted message, S is the source, and D is the destination of the message. This message contains the allocated pairwise key K_i distributed by Algorithms 1 and Algorithm 2.

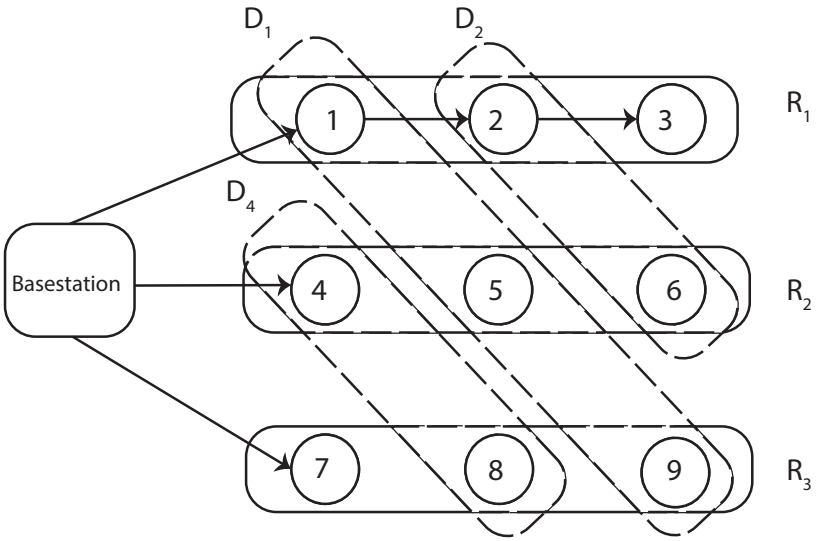


Figure 4.9: Dynamic pairwise key distribution.

The steps for dynamically distributing keys to the sensors are as follows:

(a) The bs calculates $M = E_{K_3}(K_2||ID_3)$, appends the IDs of S (bs) and D (node 3) to the messsage, and forwards M to sensor 1.

$$BS \rightarrow \text{Sensor } 1 : M_{(ID_{bs}, ID_3)}$$

(b) Sensor 1 will not be able to decrypt M as it does not know E_{K_3}. Sensor 1 can determine the source (bs) and destination (sensor 3) from the IDs appended to M. Thus sensor 1 will update the source ID to become ID_1:

$$\text{Sensor } 1 \rightarrow \text{Sensor } 2 : M_{(ID_1, ID_3)}$$

(c) Sensor 2 will not be able to decrypt M as it does not know E_{K_3}. Sensor 2 can determine the source (sensor 1) and destination (sensor 3) from the IDs appended to M. Thus sensor 2 will update the source ID to become ID_2

$$\text{Sensor } 2 \rightarrow \text{Sensor } 3 : M_{(ID_2, ID_3)}$$

(d) Upon receiving M, sensor 3 will be able to decrypt the message using its master key K_3 and retrieve the required pairwise key.

We could take the advantage of the technique called *flooding*, where each M is forwarded to the neighbour except for the source. When a sensor can decrypt M, the sensor will not forward the data. Flooding is a fast technique but the same M will be generated by different sensors causing data redundancy. However, it is possible to limit the usage of flooding for pairwise key distribution only.

4.7 Direct Key Establishment among Sensors

After the initial predistribution step, if a sensor node x wants to share a key with sensor node y, node x can easily determine if they belong to the same row group via two methods:

Method one: This method requires the establishment of a secure communication as described in Section 4.3 before the actual data exchange. Next, sensors exchange their group array to find out common group values, as discussed in Section 4.2. If the first value of the array matches, then the sensors belong to the same row and if the second value matches, then the sensors belong to the same diagonal.

For example, sensor 2 and 6 exchange their group arrays which are [1,2] and [2,2] respectively. The first array value does not match but the second value matches. Thus the source sensor 2 knows that the destination node is on the same diagonal. Hence, a direct connection can be established.

Method two: This method does not require that a secure connection be established

before the actual data exchange. The source sensor x can compute the row value of the destination sensor y by taking the ceiling value of $\lceil \frac{ID(y)}{m} \rceil$. This can be expressed as follows:

$$\text{if } \lceil \frac{ID(x)}{m} \rceil = \lceil \frac{ID(y)}{m} \rceil \text{ then } x, y \in R_i \tag{4.5}$$

For instance, in Figure 4.1 if sensor 1 wants to communicate with sensor 2 then sensor 1 computes the following values:

$$\lceil \frac{1}{3} \rceil = 0.333 = 1$$
$$\lceil \frac{2}{3} \rceil = 0.666 = 1$$

Since the values match, sensor 1 and 2 are in same row group. Alternatively, if sensor 1 wants to communicate with sensor 7 then sensor 1 computes:

$$\lceil \frac{1}{3} \rceil = 0.333 = 1$$
$$\lceil \frac{7}{3} \rceil = 2.333 = 3$$

Thus, sensors 1 and 7 are not in same row group.

For the sensors which belong to diagonal groups, if the difference of their IDs is $m+1$ or its multiple then they belong to the same diagonal group. This can be expressed as follows:

$$\text{if } |ID(x) - ID(y)| = m+1 \text{ then } x, y \in D_i \tag{4.6}$$

For instance, in Figure 4.1 if sensor 1 wants to communicate with sensor 5 or sensor 9, then

$$|1-5| = 4 = m+1$$
$$and |1-9| = 8 = \text{Multiple of (m + 1)}$$

Thus, sensors 1, 5 and 9 are in the same diagonal group.

Also if x wants to know which diagonal (D_i) x belongs to then the sensor calculates $|i - ID(\mathrm{x})|mod(m+1)$. If the value is 0 then x belongs to that diagonal. For example if sensor 8 wants to know whether it belongs to D_4 it simply calculates $|4-8|mod(m+1)$ = 0. Thus sensor 8 belongs to D_4.

4.8 Pathkey Establishment among Sensors

If there is no common key to establishing direct communication, then sensors must find keys of other sensors which lie between the source and the destination. These are called pathkeys. The participating sensors will establish direct keys between the intermediate sensors using one of the method described in Section 4.6. The steps are as follows:

1. Sensors in the same diagonal groups (D_a) require the minimum number of keys for encryption. Thus, the source sensor node x calculates whether the destination sensor node y is in the same diagonal group. If y belongs to the same diagonal group, then x forwards the data to sensor y using the common diagonal group D_a.

2. Sensors in the same row groups (R_a) also require the minimum number of keys for encryption. If step 1 fails, then the source sensor node x calculates whether the destination sensor node y is in the same row group. If y belongs to the same row group, then x forwards the data to sensor y using the common row group R_a.

3. If the destination sensor y does not belong to the same row group or diagonal group of the source sensor x, then x looks for a node z in the neighbourhood which can reach the row group of y. To do this, x sends a message containing the ID of the destination sensor to z. z tries to establish a direct key with the destination sensor y using one of the methods described in Section 4.6. If a match is found, then z takes the data from x to the destination y.

4. If all of the above fails, the source node x sends a message containing the ID of the destination sensor to the next sensor of own diagonal (this enables the data to flow to a new row). That sensor then follows Step 1 through 4 until the data reaches the destination.

5. If multiple paths are found with the same number of hops, the source will choose the path which will require minimum number of keys for encryption.

As an example let us apply this to the network as shown in Figure 4.1 where $m = 3$:

(a) Node 1 wants to communicate with node 6. Here, the group arrays of node 1 and 6 are $\{1,1\}$ and $\{2,2\}$ respectively. Using one of the methods described in Section 4.6 node 1 can identify that node 1 and 6 do not have any common row groups or diagonal groups.

(b) Then, node 1 communicates with the neighbouring nodes and retrieves their group arrays:

Neighbour sensor	Group array
2	$\{1,2\}$
5	$\{2,1\}$

(c) From the arrays, node 1 can easily identify that node 5 is in row 2, which is also the row of the destination sensor 6, $\lceil \frac{6}{3} \rceil = 2$. The final path is:

$$\text{Path 1: Sensor 1} \xrightarrow{D_1} \text{Sensor 5} \xrightarrow{R_1} \text{Sensor 6}$$

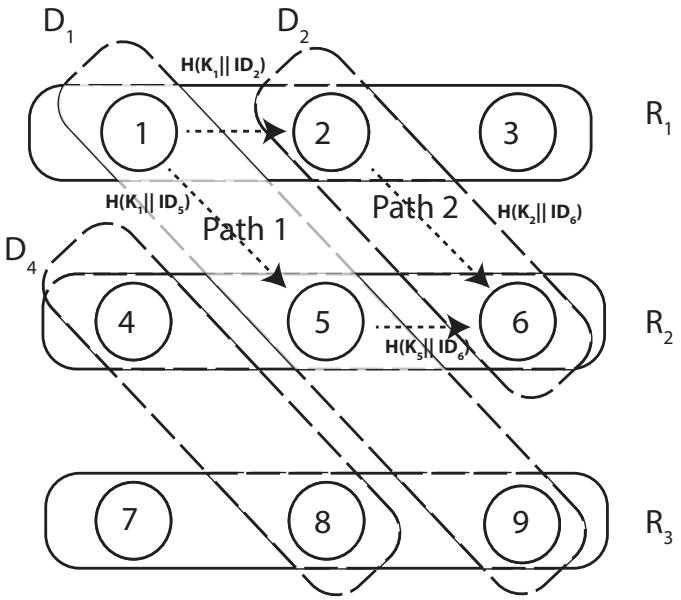


Figure 4.10: Pathkey establishment for diagonal-based grouping.

However for diagonal$_{min}$ grouping, there is only one possible path illustrated in Figure 4.11.

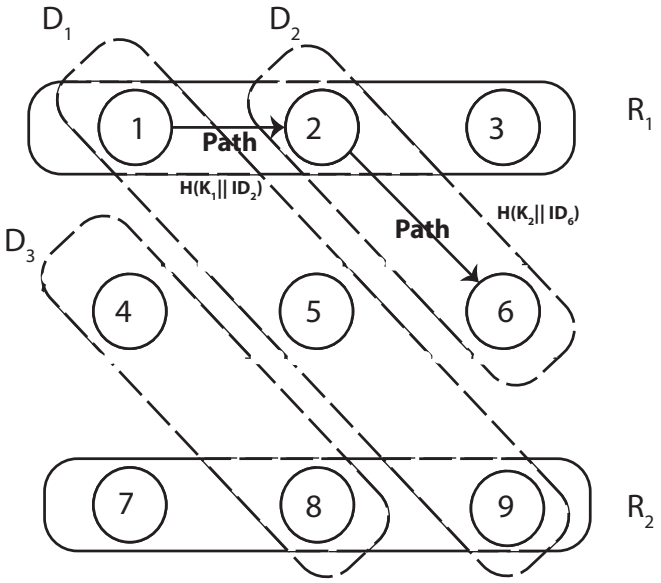


Figure 4.11: Pathkey establishment for diagonal$_{min}$ grouping.

For another example, Figure 4.1 is used,

(a) Node 1 wants to communicate with node 7. Here, the group arrays of nodes 1 and 7 are {1,1} and {3,0} respectively. Using one of the methods described in Section 4.6

node 1 can identify that node 1 and 7 do not have any common row groups or diagonal groups.

(b) Node 1 then communicates with the neighbouring nodes and retrieves their group arrays:

Neighbour sensor	Group array
2	$\{1,2\}$
5	$\{2,1\}$

(c) Node 1 forwards data along with the ID of destination sensor 7 to node 5. Because node 5 belongs to row 2 which is a neighbour of the destination row, $\lceil \frac{7}{3} \rceil = 3$.

(d) Sensor 5 communicates with the neighbouring nodes and retrieves their respective rows and columns:

Neighbour sensor	Group array
6	$\{2,2\}$
9	$\{3,1\}$

Now, node 5 forwards data to node 9 of row 3 which is in the same row of the destination sensor 7, $\lceil \frac{7}{3} \rceil = 3$.

(e) Sensor 9 (group array $\{3,1\}$) forwards the data to sensor 8 (group array $\{3,4\}$) (first element of group array matches).

(f) Finally, sensor 8 (group array $\{3,4\}$) delivers the data to destination sensor 7 (group array $\{3,0\}$) (first element of group array matches).

The final path is:

$$\text{Path 1: Sensor 1} \xrightarrow{D_1} \text{Sensor 5} \xrightarrow{D_1} \text{Sensor 9} \xrightarrow{R_3} \text{Sensor 8} \xrightarrow{R_3} \text{Sensor 7}$$

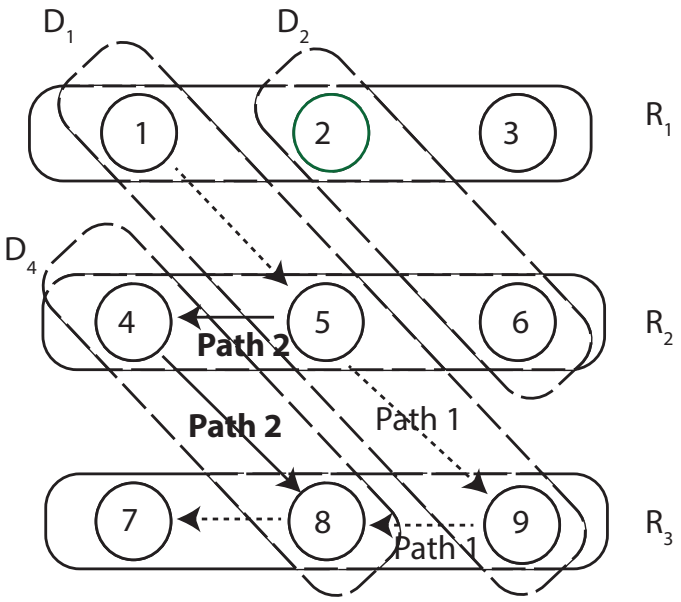


Figure 4.12: Pathkey establishment for diagonal-based grouping.

For example, if sensor 1 wants to communicate with sensor 3 the possible paths are:

$$\text{Path 1: Sensor 1} \xrightarrow{R_1} \text{Sensor 2} \xrightarrow{R_1} \text{Sensor 3}$$

$$\text{Path 2: Sensor 1} \xrightarrow{D_1} \text{Sensor 5} \xrightarrow{R_2} \text{Sensor 6} \xrightarrow{D_2} \text{Sensor 2} \xrightarrow{R_1} \text{Sensor 3}$$

From the hash key distribution as illustrated in Figure 4.6 we can see that node 1 is already distributed with the hash key of node 3. Thus, path 1 will require only one key to encrypt while path 2 will require four hash keys.

We have also implemented the network orientation used in [1] as illustrated in Figure 4.13 where $m = 4$:

(a) Node 1 wants to communicate with node 8. Here, the group arrays of node 1 and 8 are $\{1,1\}$ and $\{2,3\}$ respectively. Using one of the methods described in Section 4.6 node 1 can identify that node 1 and 8 do not have any common row groups or diagonal groups.

(b) Node 1 then communicates with the neighbouring nodes and retrieves their group arrays (here, sensor 2 group array: $\{1,2\}$ and sensor 6 group array: $\{2,1\}$) and calculates their respective row groups and diagonal groups to reach the destination sensor 8 (group array: $\{2,3\}$).

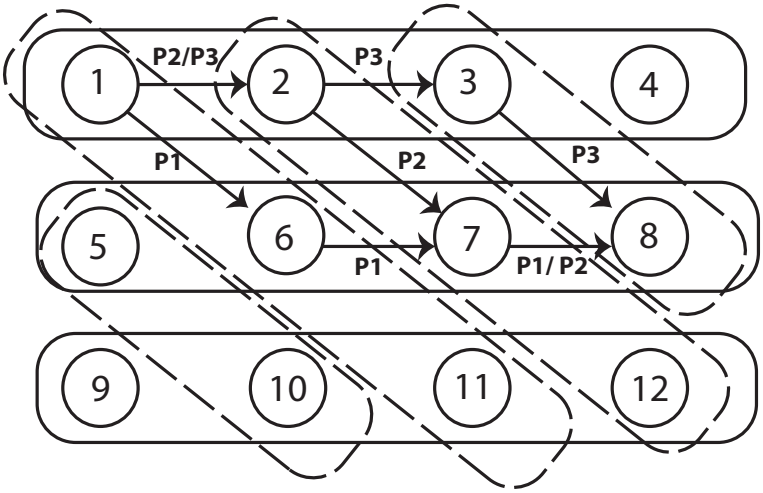


Figure 4.13: Pathkey establishment for diagonal-based grouping using Liu's example [1].

(c) As sensor 6 resides in the same row as sensor 8, $\lceil \frac{6}{4} \rceil = 2$, the source sensor 1 will forward the data to sensor 6. Then sensor 6 will forward the data to sensor 7 which in turn will send the data to destination 8. The path is as follows:

$$\text{Path 1: Sensor 1} \xrightarrow{D_1} \text{Sensor 6} \xrightarrow{R_2} \text{Sensor 7} \xrightarrow{R_2} \text{Sensor 8} \xrightarrow{R_3} \text{Sensor 7}$$

Consider this example where sensor 14 wants to communicate with sensor 22 as shown in Figure 4.14. The group arrays of nodes 14 and 22 are $\{3,2\}$ and $\{5,16\}$ respectively and $m = 5$.

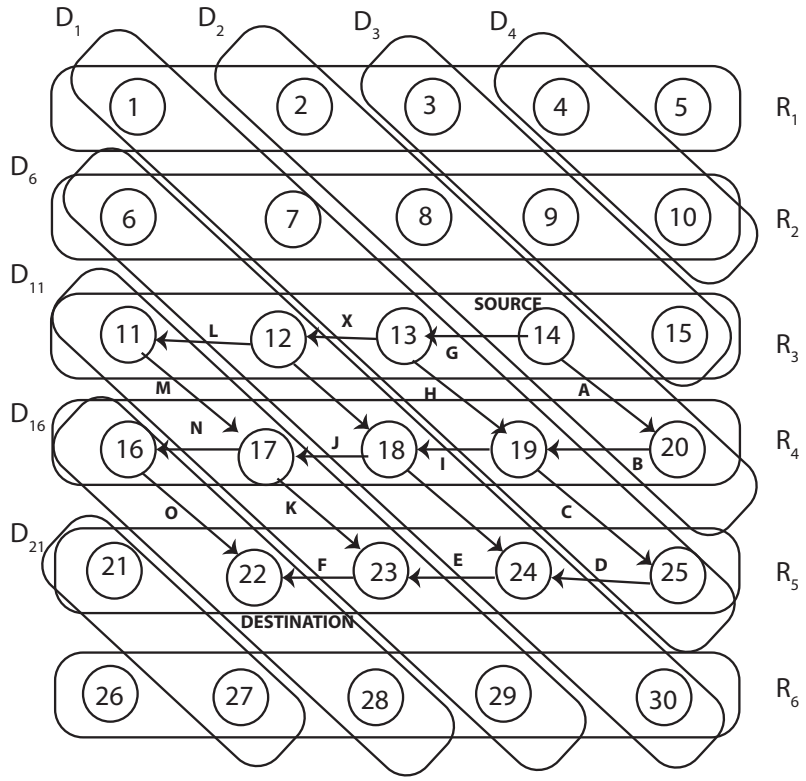


Figure 4.14: Pathkey establishment for diagonal-based grouping for 6 $\times$5 network.

The path will be:

$$\text{Path 1: Sensor 14} \xrightarrow{R_3} \text{Sensor 13} \xrightarrow{R_3} \text{Sensor 12} \xrightarrow{R_3} \text{Sensor 11}$$

$$\xrightarrow{D_{11}} \text{Sensor 17} \xrightarrow{D_{11}} \text{Sensor 23} \xrightarrow{R_5} \text{Sensor 22(Required keys: 3)}$$

The path can be rewritten in the following way as illustrated Figure 4.14:

$$G \rightarrow X \rightarrow L \rightarrow M \rightarrow K \rightarrow F$$

There are many possible paths from source to destination. Figure 4.15 shows a simple tree diagram illustrating all the possible paths.

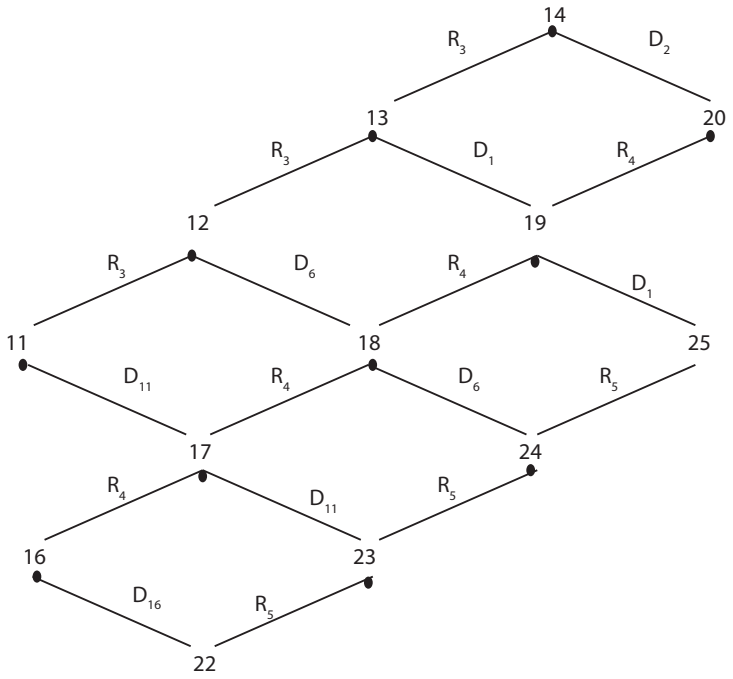


Figure 4.15: Tree diagram of a 6 $\times$5 network for pathkey establishment.

Three possible paths from node 14 to node 22 are as follows:

$$14 \rightarrow 20 \rightarrow 19 \rightarrow 25 \rightarrow 24 \rightarrow 23 \rightarrow 22$$

$$14 \rightarrow 13 \rightarrow 12 \rightarrow 11 \rightarrow 17 \rightarrow 16 \rightarrow 22$$

$$14 \rightarrow 20 \rightarrow 19 \rightarrow 18 \rightarrow 17 \rightarrow 23 \rightarrow 22$$

4.9 Pathkey in case of Node Compromise

Naturally, during network operation some of the sensor nodes will become inactive due to low battery, or be compromised by an adversary.

Figure 4.12 illustrates that node 1 has already established communication with node 6

via node 5 using path 1. But if sensor 5 is compromised then data sent to sensor 5 will fail. In this situation sensor 2 re-evaluates pathkey establishment steps described in section 4.7 with the neighbour (node 2) and calculates $|2-6| = 4 = m+1$. Thus, the destination sensor is on the same diagonal. Alternatively, if sensor 2 and sensor 6 compare their group arrays, {1,2} and {2,2} respectively, they will find a common diagonal, {2}. Then sensor 1 forwards data to sensor 2.

$$\text{Path 2: Sensor 1} \xrightarrow{R_1} \text{Sensor 2} \xrightarrow{D_2} \text{Sensor 6}$$

In a different scenario also using Figure 4.12, where the source is sensor 1 with group array {1,1} and the destination is sensor 7 with group array {2,2}, if sensor 9 of path 1 is compromised then node 5 will communicate with the neighbouring nodes and retrieves their respective rows and columns:

Neighbour sensor	Group array
1	$\{1,1\}$
4	$\{2,4\}$
6	$\{2,2\}$
9	$\{3,1\}$

From these arrays, node 1 can identify that node 4 is in row 2 which is a neighbour to the row of the destination sensor 7, $\lceil \frac{7}{3} \rceil = 3$.

$$\text{Path 2: Sensor 1} \xrightarrow{D_1} \text{Sensor 5} \xrightarrow{R_2} \text{Sensor 4} \xrightarrow{D_4} \text{Sensor 8} \xrightarrow{R_3} \text{Sensor 7}$$

Finally, as shown in Figure 4.16 where the source is sensor 1 with group array {1,1} and the destination is sensor 8 with group array {2,3}, if sensor 6 of path 1 is compromised then sensor 2 selects the neighbouring nodes (here, sensor 3 and sensor 7) and calculates their respective row groups and diagonal groups to reach the destination sensor 8.

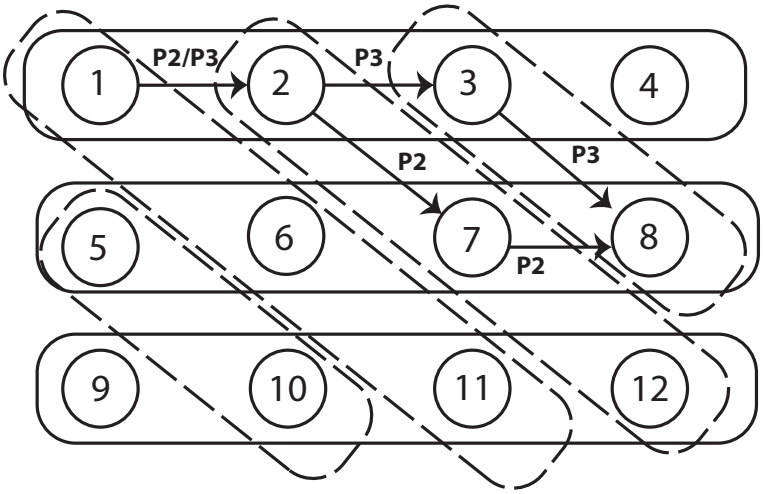


Figure 4.16: Pathkey establishment for diagonal-based grouping using Liu's example [1].

(a) As sensor 7 resides in the same row as sensor 8, $\lceil \frac{7}{4} \rceil = 2$, the source sensor 2 will forward the data to sensor 7. Sensor 7 will forward the data to destination sensor 8. The path is as follows:

$$\text{Path 2: Sensor 1} \xrightarrow{R_1} \text{Sensor 2} \xrightarrow{D_2} \text{Sensor 7} \xrightarrow{R_2} \text{Sensor 8}$$

(b) If sensor 7 is also compromised, then sensor 2 chooses sensor 3 from its own row which has a common diagonal with the destination sensor 8. The path is as follows:

$$\text{Path 3: Sensor 1} \xrightarrow{R_1} \text{Sensor 2} \xrightarrow{R_1} \text{Sensor 3} \xrightarrow{D_3} \text{Sensor 8}$$

Thus, if sensor 1 wants to communicate with sensor 8 the possible paths are:

Path 1: Sensor 1 $\xrightarrow{D_1}$ Sensor 6 $\xrightarrow{R_2}$ Sensor 7 $\xrightarrow{R_2}$ Sensor 8 (Required keys: 2)

Path 2: Sensor 1 $\xrightarrow{R_1}$ Sensor 2 $\xrightarrow{D_2}$ Sensor 7 $\xrightarrow{R_2}$ Sensor 8 (Required keys: 3)

Path 3: Sensor 1 $\xrightarrow{R_1}$ Sensor 2 $\xrightarrow{R_1}$ Sensor 3 $\xrightarrow{D_3}$ Sensor 8 (Required keys: 2)

All of the paths above require three hops from source to destination. However, as shown in Figure 4.8 for path 1, node 6 and node 8 already have a pairwise key in common. Thus, the pairwise key of node 7 is not required here. Path 2 also does not need the pairwise key of node 2. Unlike paths 1 and 2, path 3 requires three pairwise keys. This allows the proposed algorithm to choose either path 1 or 3 over path 2.

Chapter 5

Computational Results and Performance Analysis

In this chapter, the computational results and performance of the proposed diagonal-based grouping are presented based on different network orientations. The parameters considered are key storage and pathkey length. Our results are also compared with previously conducted research on group based key distribution done by Liu et al.[1]. This is henceforth referred to as Liu's grouping.

5.1 Network Orientation

The proposed grouping framework has been implemented in three different orientations of the network. The purpose was to see if the performance varies based on the orientation of the network. There appears to be significant evidence that the performance does vary with the network orientation. Three different orientations were considered:

1. A network with equal rows and columns where the number of rows (n) and sensors in each row (m) are the same, that is, $n = m$. Example: 3×3, 4×4, 5×5, or 6×6 networks.

2. A network with fewer rows than columns, that is, where $n < m$. Example: 3×4, 3×7, 4×5, 4×11, 5×6, or 6×7 networks.

3. A network with more rows than columns, that is, where $n > m$. Example: 4×3, 7×4, 5×4, 6×5, or 12×6 networks.

5.2 Key Storage

The goal is to find a layout which will allow sensors to store the least number of keys without compromising the security integrity of the network. First, we examine the number of keys that are required to be stored for each group design. We then calculate the ratio for each sensor.

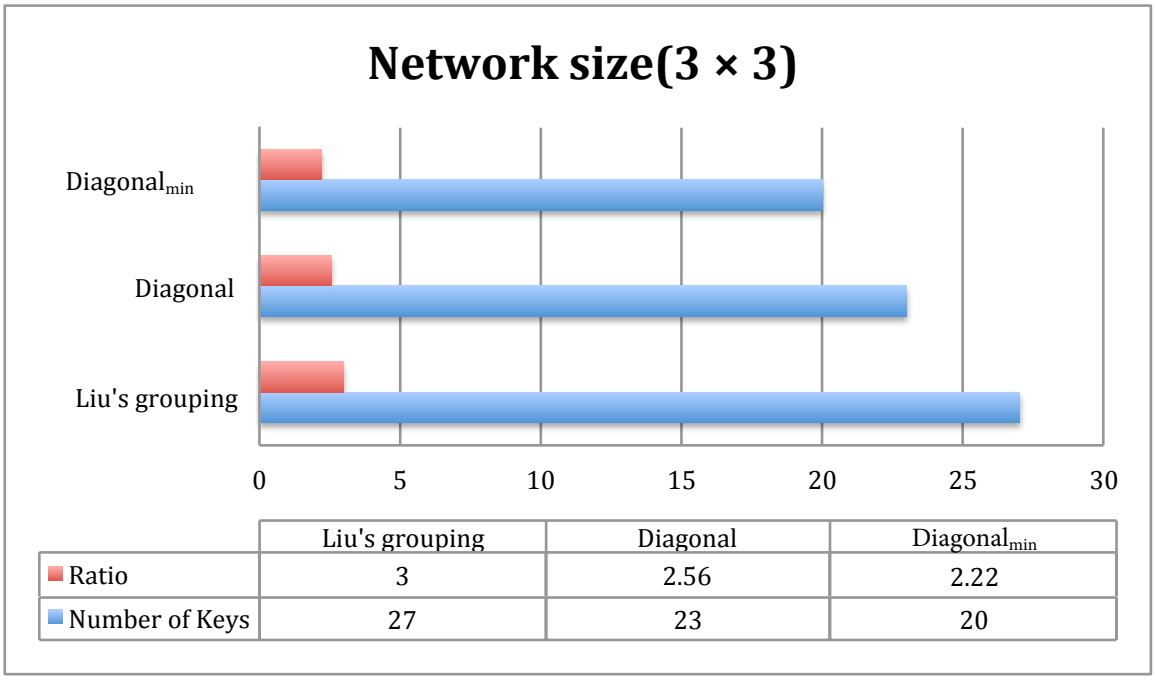
$$ratio = \frac{\text{Total Number of Keys}}{\text{Total number of sensors}}$$

This ratio indicates the average key storage load on each sensor of a network.

We start with a small network (3×3) consisting of 9 sensors. As we increase the number of sensors, Figure 5.5 shows that our proposed diagonal-based grouping requires fewer keys to be stored as the number of sensors increases. For $\text{diagonal}_{\text{min}}$ grouping the required keys are much fewer but in the next section we see that there is a trade-off with the pathkey length to achieve this lower number.

5.2.1 Networks with equal rows and columns ($n = m$)

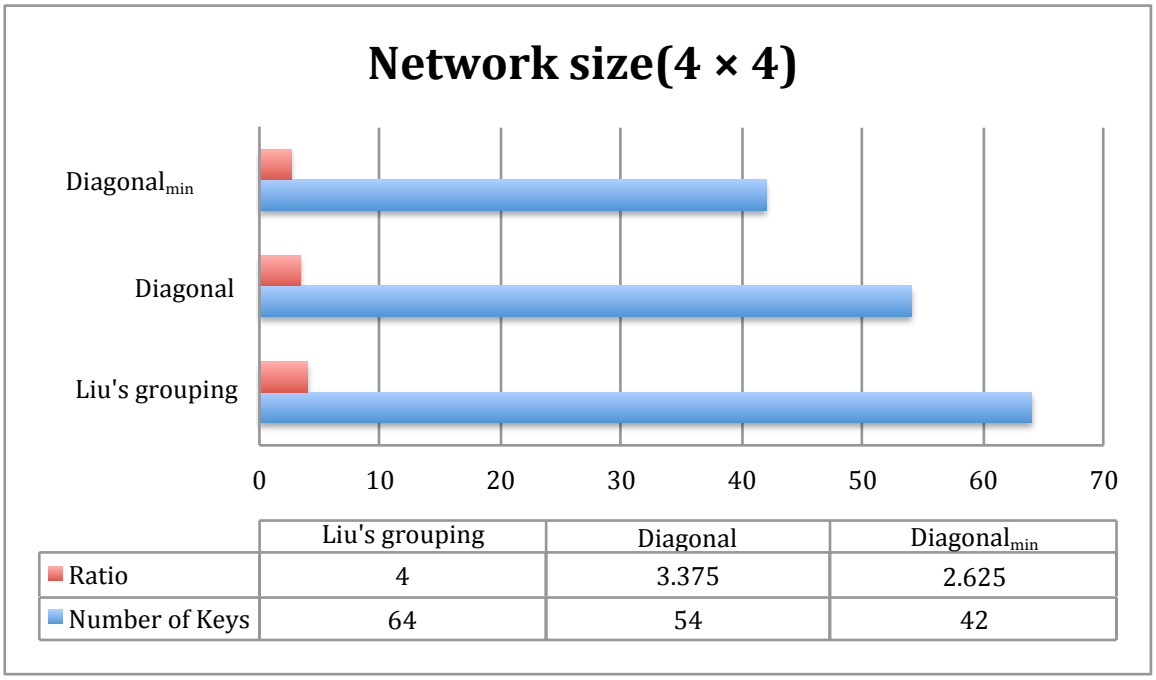
First we consider networks where the number of rows and the number of columns is equal. This results in a square-shaped orientation.



	Liu's grouping	Diagonal	Diagonal$_{min}$
Ratio	3	2.56	2.22
Number of Keys	27	23	20

Figure 5.1: Total number of keys and key-to-sensor ratio for a 3×3 network.

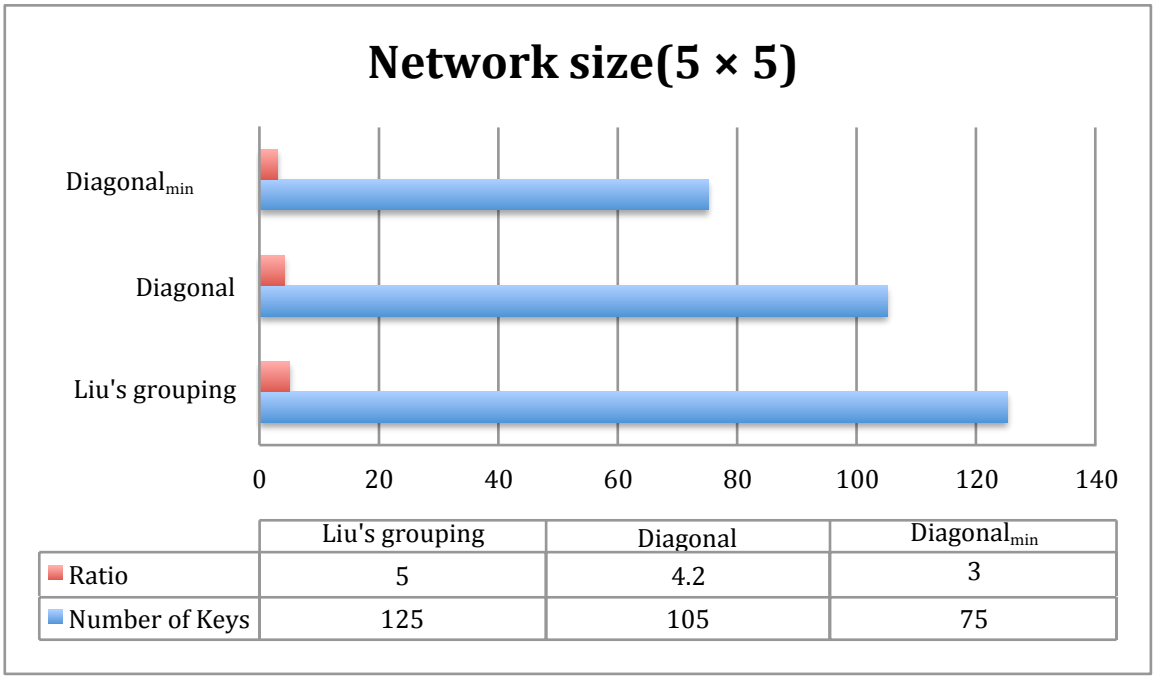
From Figure 5.1 we can see that in a 3×3 network, Liu's grouping requires 27 keys to be stored with a ratio of 3 keys per sensor. Diagonal-based grouping requires 23 keys to be stored with a ratio of 2.56 while diagonal$_{min}$ grouping requires 20 keys to be stored with a ratio of 2.22.



	Liu's grouping	Diagonal	Diagonal$_{min}$
Ratio	4	3.375	2.625
Number of Keys	64	54	42

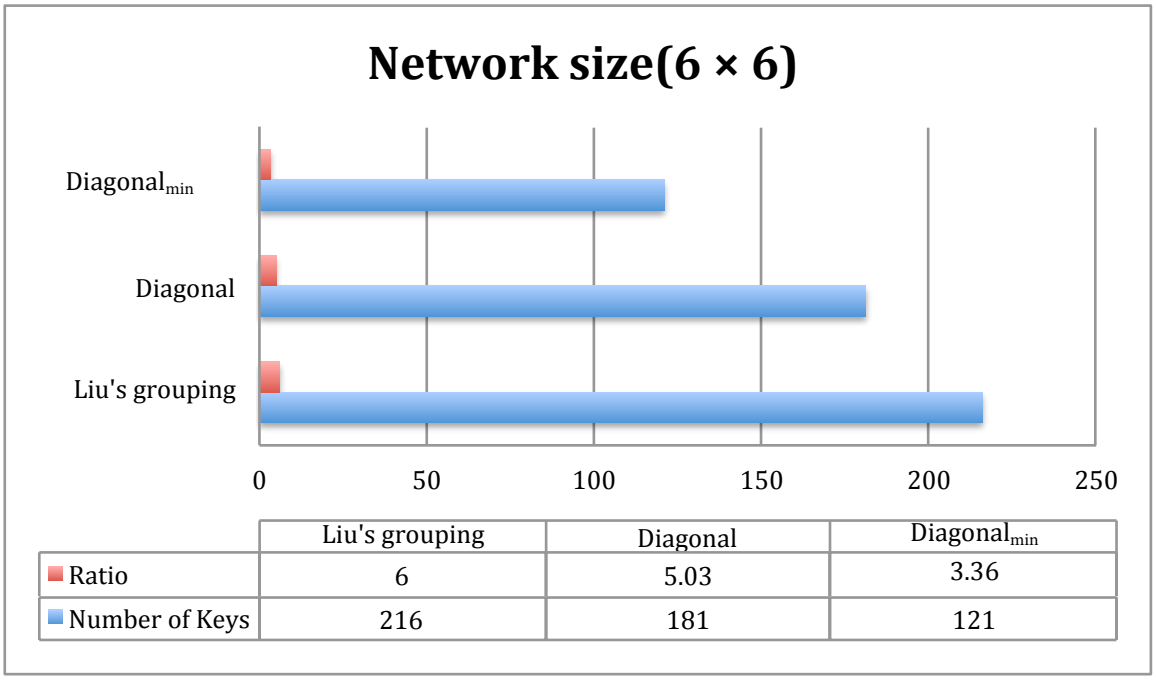
Figure 5.2: Total number of keys and key-to-sensor ratio for a 4×4 network.

For a 4×4 network, Liu's grouping requires 64 keys to be stored with a ratio of 4 keys per sensor. Diagonal-based grouping requires 54 keys to be stored with a ratio of 3.375 while diagonal$_{min}$ grouping requires 42 keys to be stored with a ratio of 2.625.



	Liu's grouping	Diagonal	Diagonal$_{min}$
Ratio	5	4.2	3
Number of Keys	125	105	75

Figure 5.3: Total number of keys and key-to-sensor ratio for a 5×5 network.



	Liu's grouping	Diagonal	Diagonal$_{min}$
Ratio	6	5.03	3.36
Number of Keys	216	181	121

Figure 5.4: Total number of keys and key-to-sensor ratio for a 6×6 network.

Figures 5.3 and 5.4 show that diagonal$_{min}$ grouping keeps the ratio around 3 even though the number of nodes increasse more than 50%.

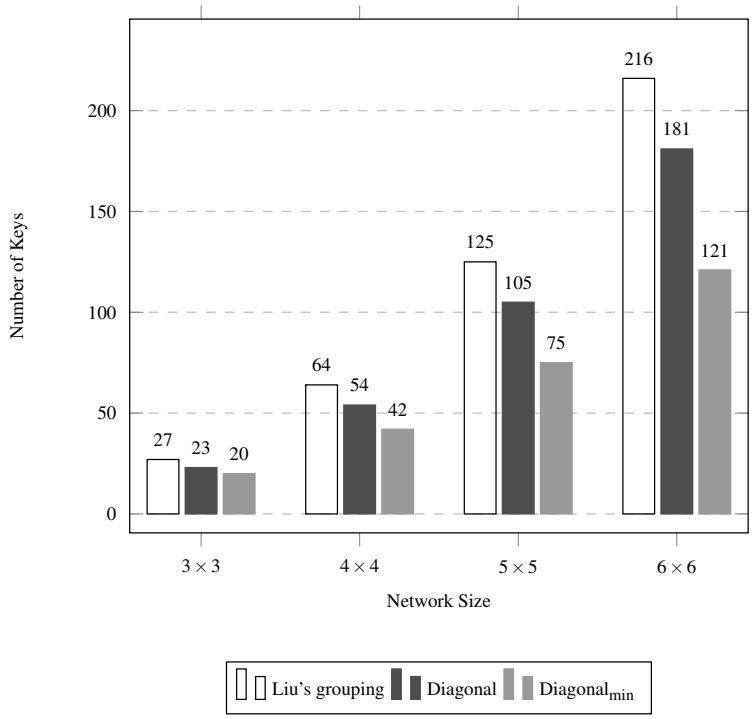


Figure 5.5: Key storage for Liu's grouping and proposed groupings where $n = m$.

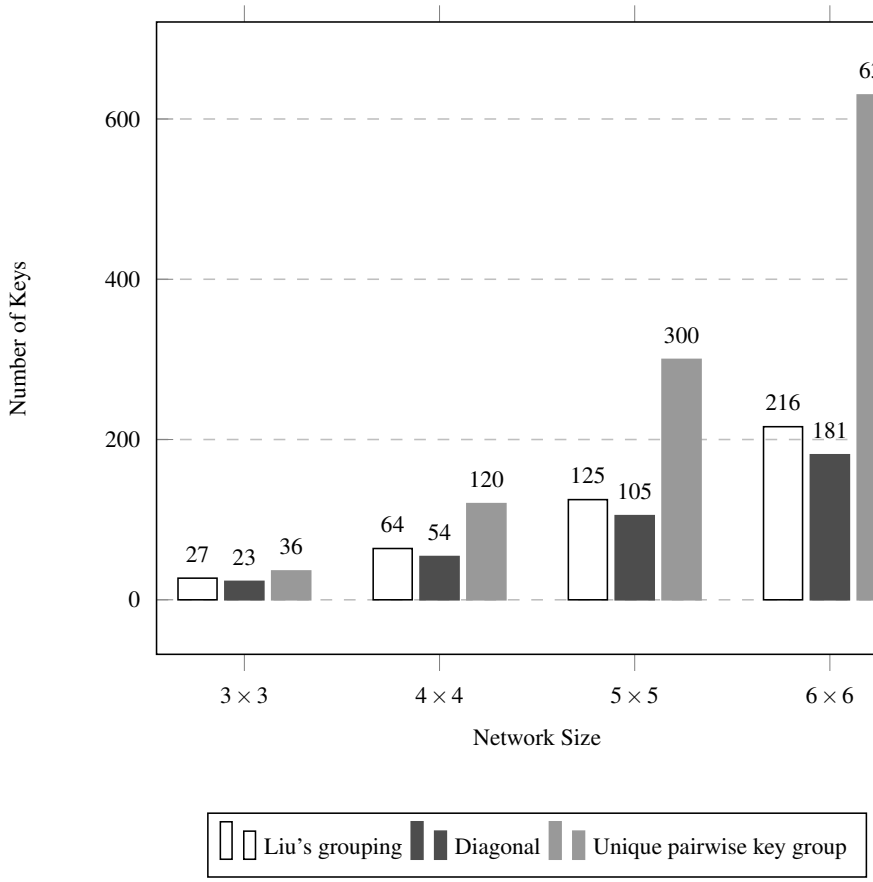


Figure 5.6: Key storage for unique pairwise key grouping and proposed groupings where $n = m$.

If we look at the pattern of required keys for these designs as shown in Figures 5.5, 5.6, 5.7, and 5.8, diagonal-based grouping is around 15% more key storage efficient than Liu's grouping. Also, the key storage efficiency for $\text{diagonal}_{\text{min}}$ continues to increase as the number of nodes increases.

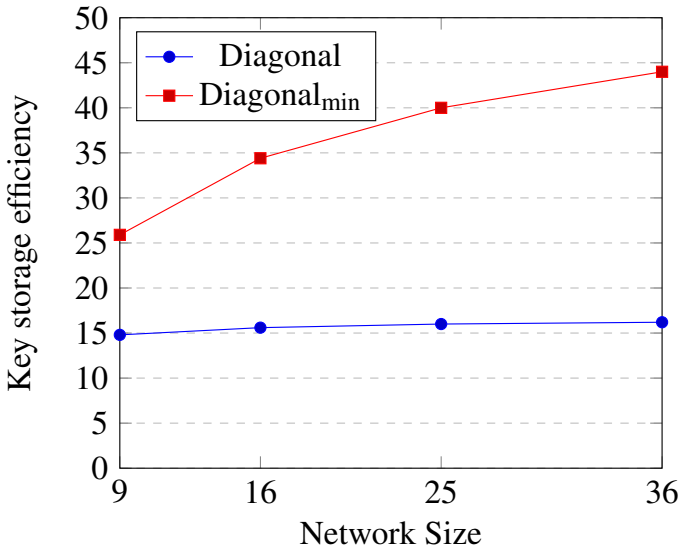


Figure 5.7: Increase in key storage efficiency where $n = m$.

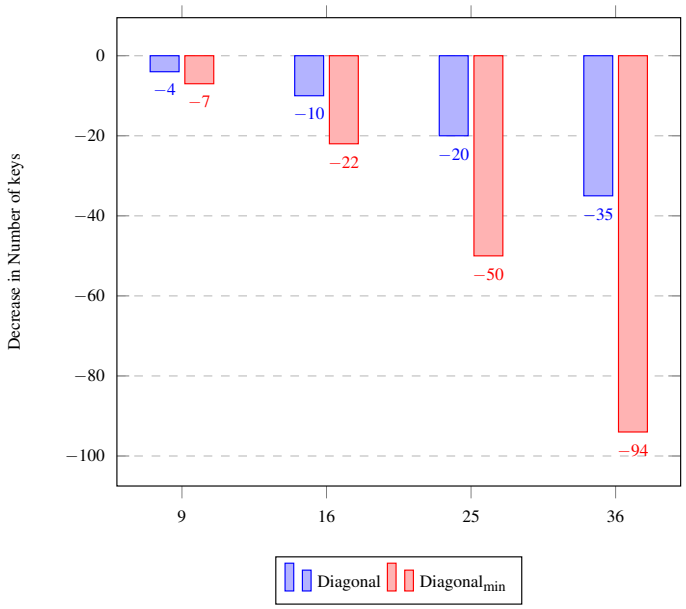
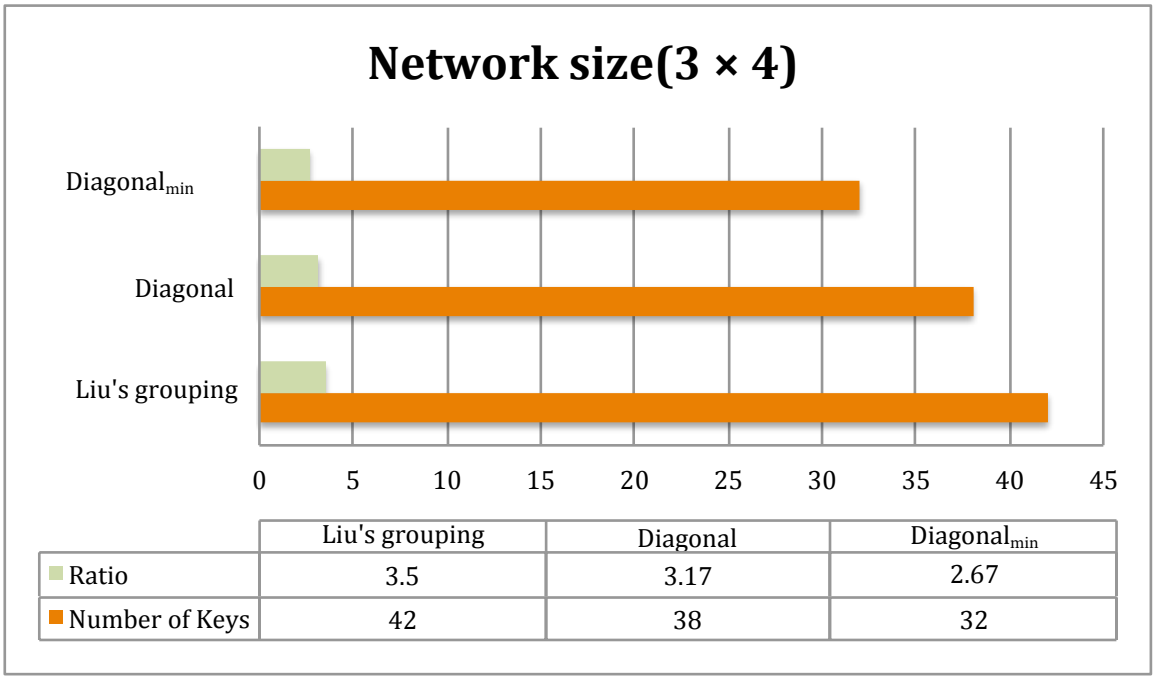


Figure 5.8: Decrease in Number of keys in terms of Liu's grouping where $n = m$.

5.2.2 Networks with fewer rows than columns ($n < m$)

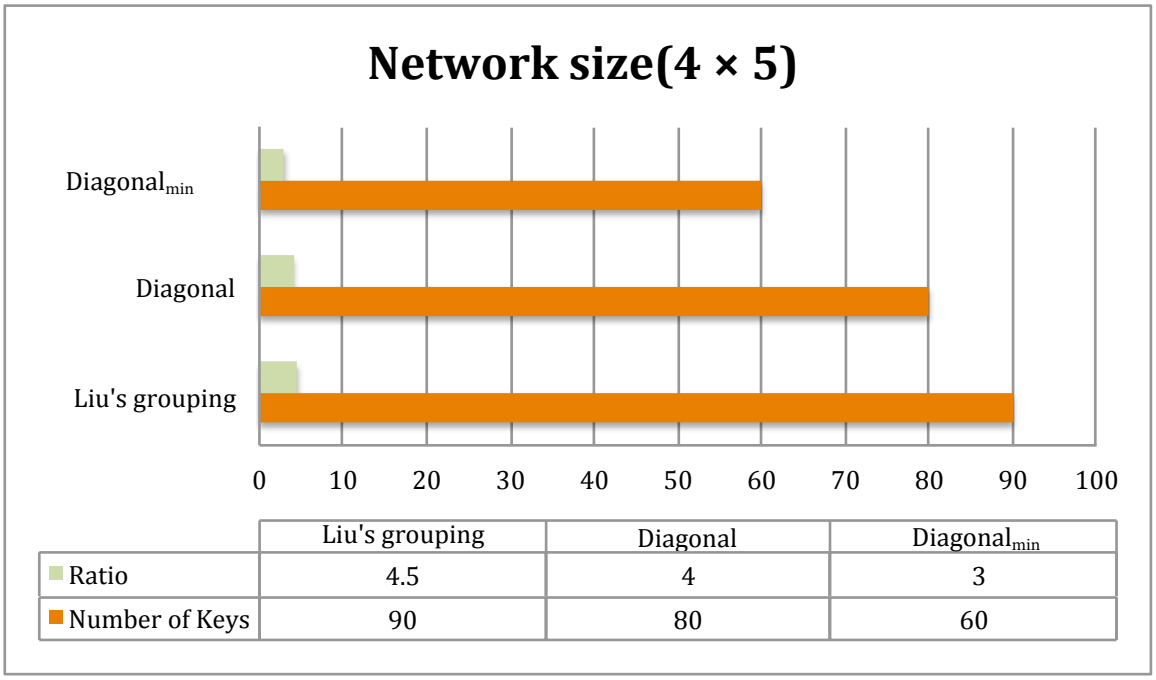
Next we consider networks with fewer rows than columns.



	Liu's grouping	Diagonal	Diagonal$_{min}$
Ratio	3.5	3.17	2.67
Number of Keys	42	38	32

Figure 5.9: Total number of keys and key-to-sensor ratio for a 3×4 network.

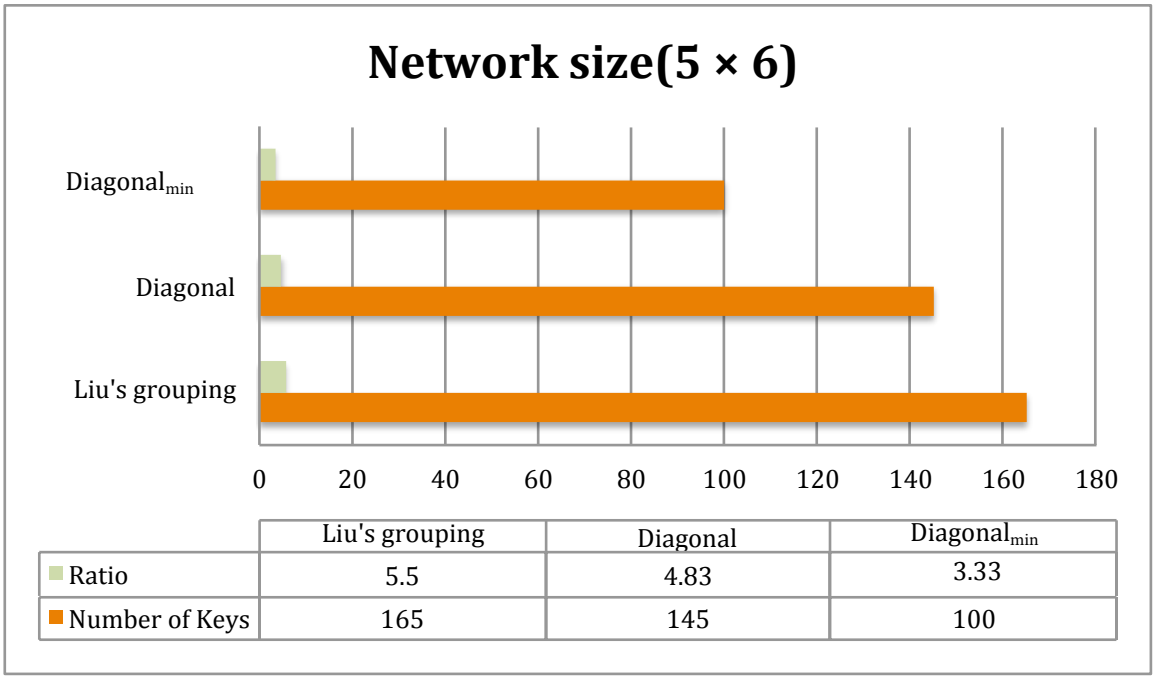
From Figure 5.9 we can see that in a 3×4 network, Liu's grouping requires 42 keys to be stored with a ratio of 3.5 keys per sensor. Diagonal-based grouping requires 38 keys to be stored with a ratio of 3.17 while diagonal$_{min}$ grouping requires 32 keys to be stored with a ratio of 2.67.



	Liu's grouping	Diagonal	Diagonal$_{min}$
Ratio	4.5	4	3
Number of Keys	90	80	60

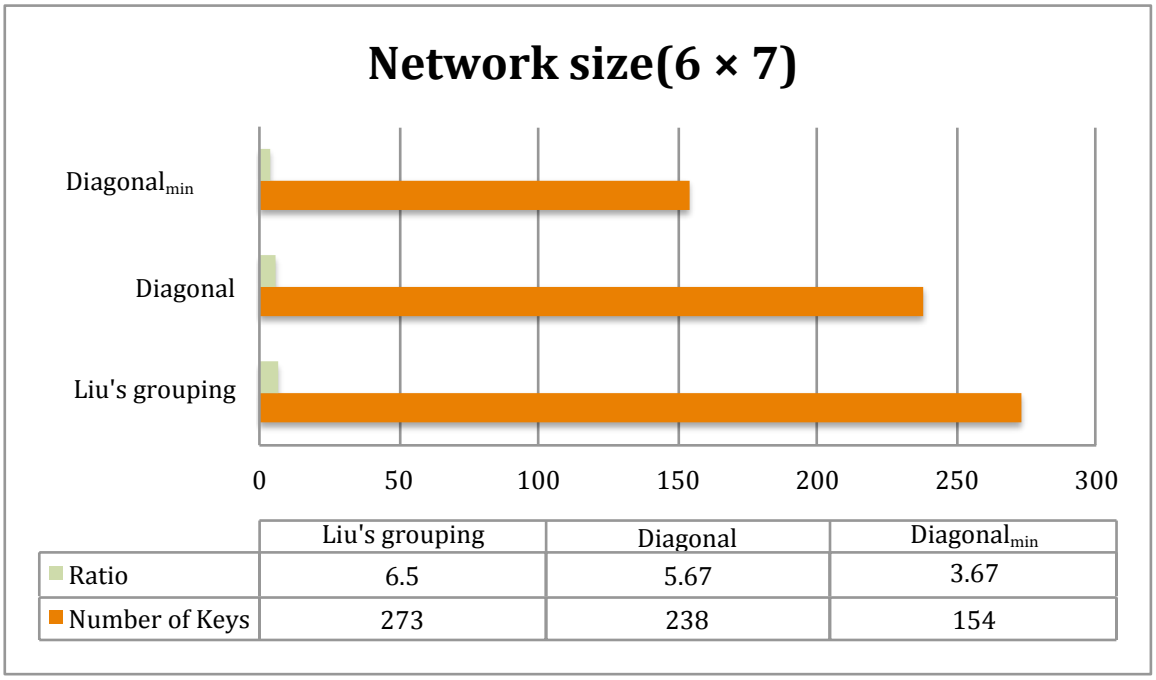
Figure 5.10: Total number of keys and key-to-sensor ratio for a 4×5 network.

For a 4×5 network, Liu's grouping requires 90 keys to be stored with a ratio of 4.5 keys per sensor. Diagonal-based grouping requires 80 keys to be stored with a ratio of 4 while diagonal$_{min}$ grouping requires 60 keys to be stored with a ratio of 3.



	Liu's grouping	Diagonal	Diagonal$_{min}$
Ratio	5.5	4.83	3.33
Number of Keys	165	145	100

Figure 5.11: Total number of keys and key-to-sensor ratio for a 5×6 network.



	Liu's grouping	Diagonal	Diagonal$_{min}$
Ratio	6.5	5.67	3.67
Number of Keys	273	238	154

Figure 5.12: Total number of keys and key-to-sensor ratio for a 6×7 network.

Figure 5.11 and 5.12 show that diagonal$_{min}$ keeps the load ratio around 3.5 even though the number of nodes increases more than 60%.

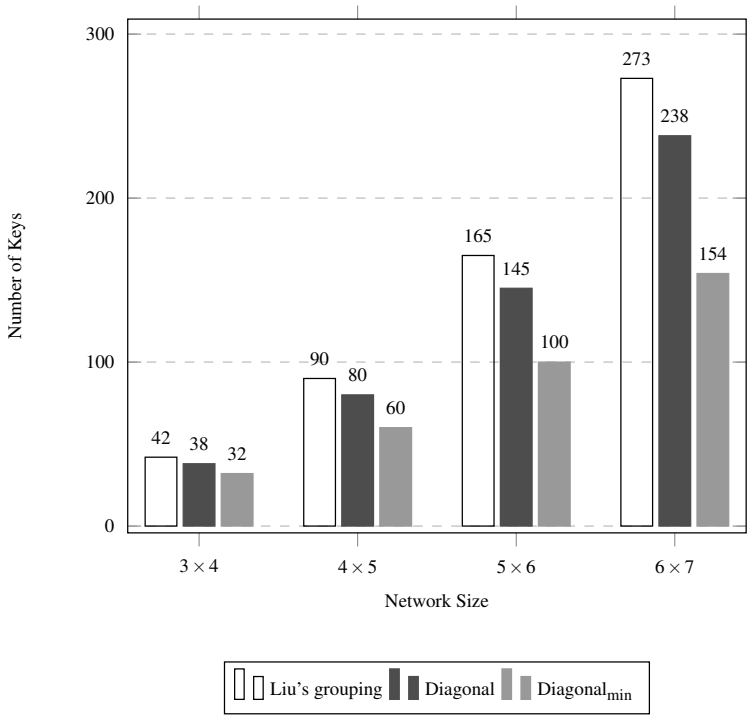


Figure 5.13: Key storage for Liu's grouping and proposed groupings where $n < m$.

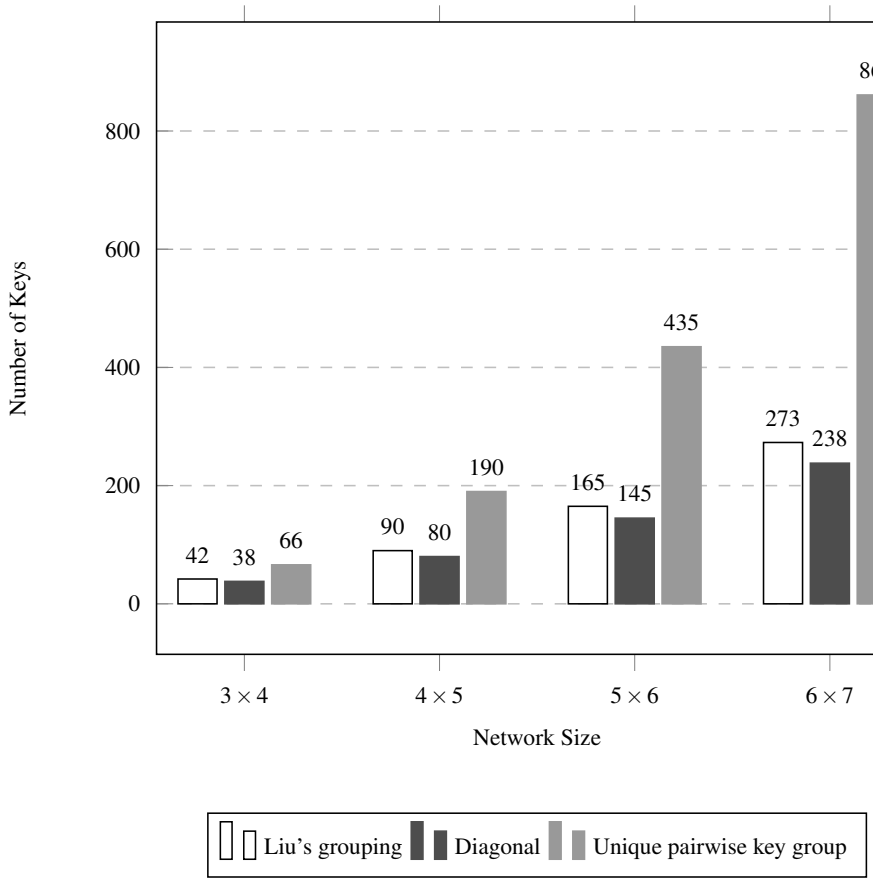


Figure 5.14: Key storage for unique pairwise key grouping and proposed groupings where $n < m$.

If we look at the pattern of required keys for the network orientations in Figures 5.13, 5.14, 5.15, and 5.16 diagonal-based grouping always requires around 12% fewer keys than Liu's grouping. At the same time, the required keys for $\text{diagonal}_{\text{min}}$ keeps decreasing as the number of nodes increases.

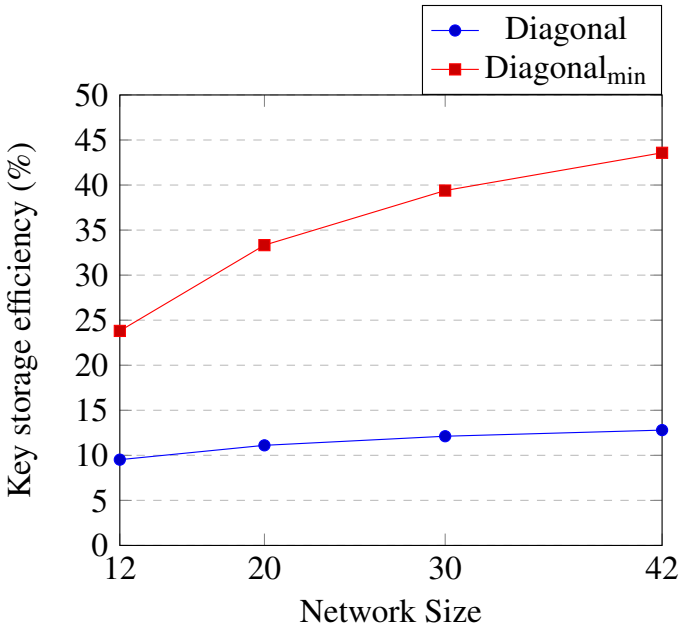


Figure 5.15: Increase in key storage efficiency where $n < m$.

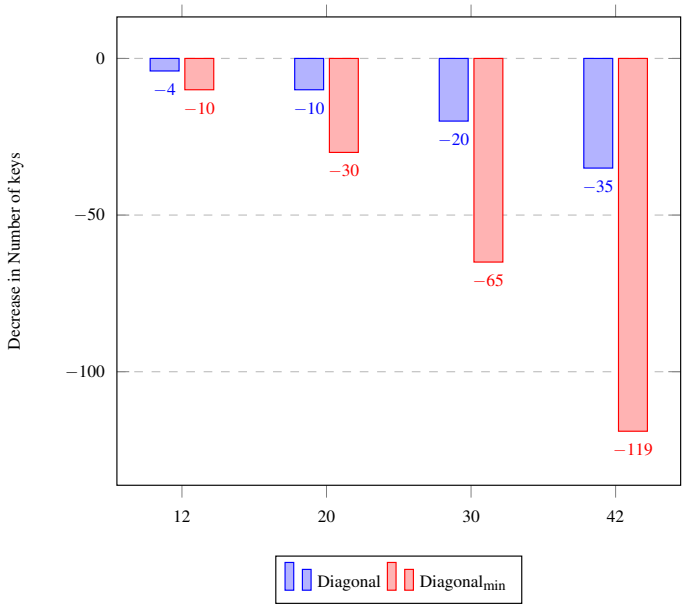
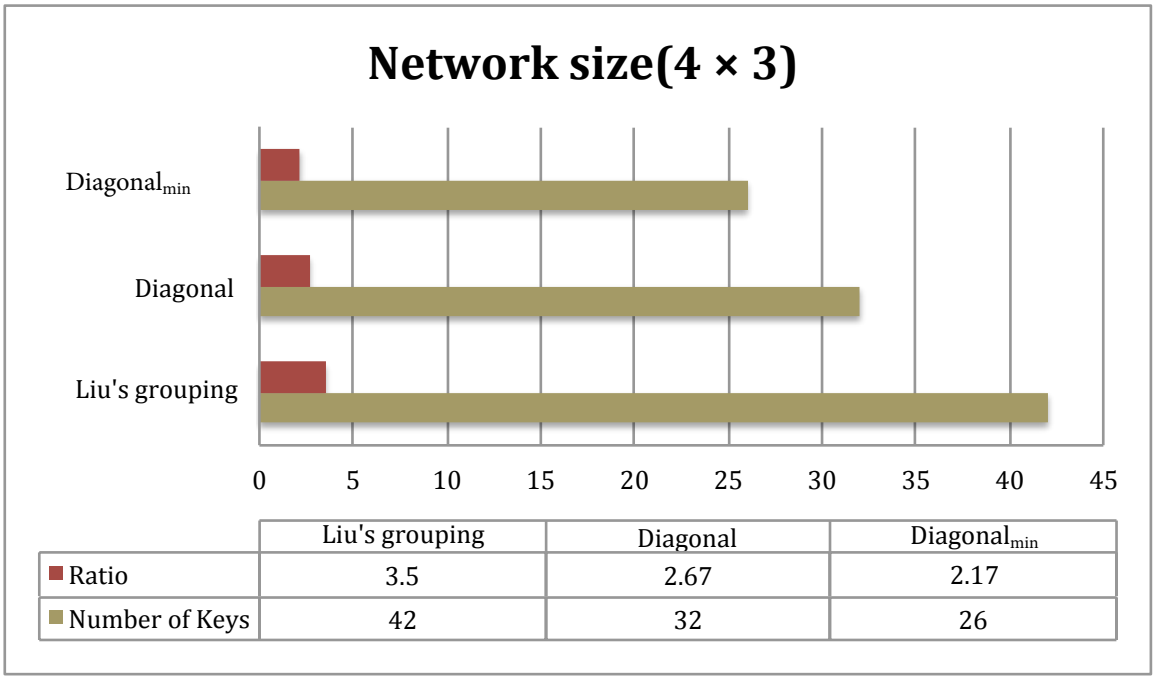


Figure 5.16: Decrease in number of keys in terms of Liu's grouping where $n < m$.

5.2.3 Networks with more rows than columns ($n > m$).

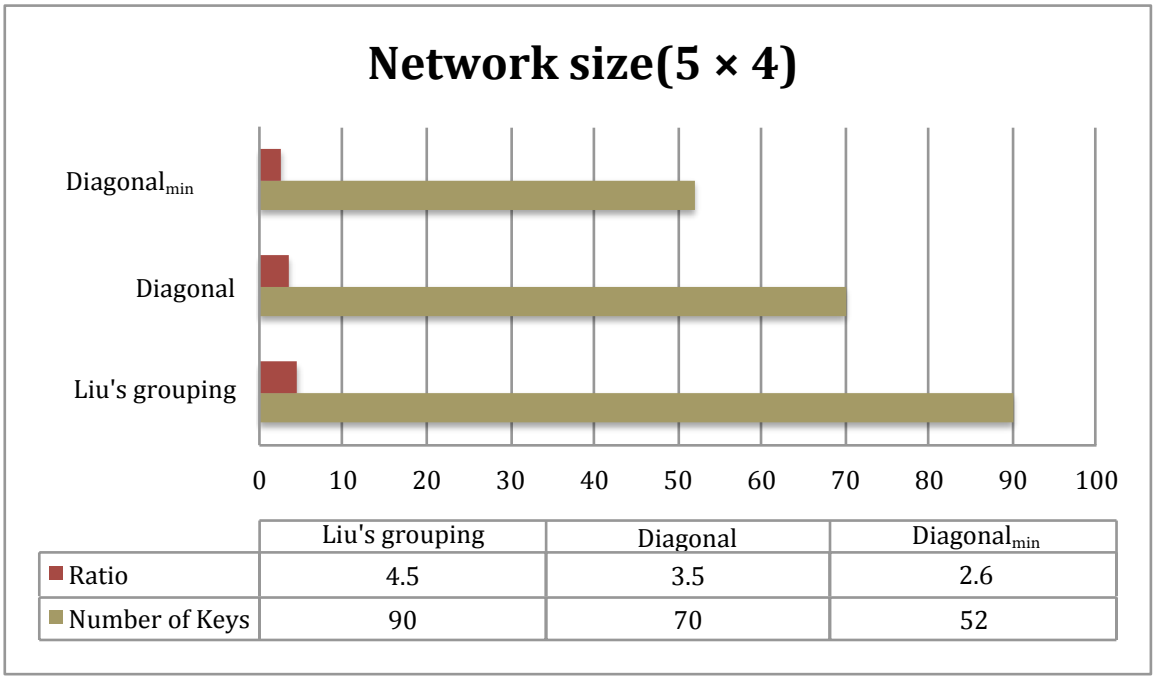
Next we consider networks with more rows than columns.



	Liu's grouping	Diagonal	Diagonal$_{min}$
Ratio	3.5	2.67	2.17
Number of Keys	42	32	26

Figure 5.17: Total number of keys and key-to-sensor ratio for a 4×3 network.

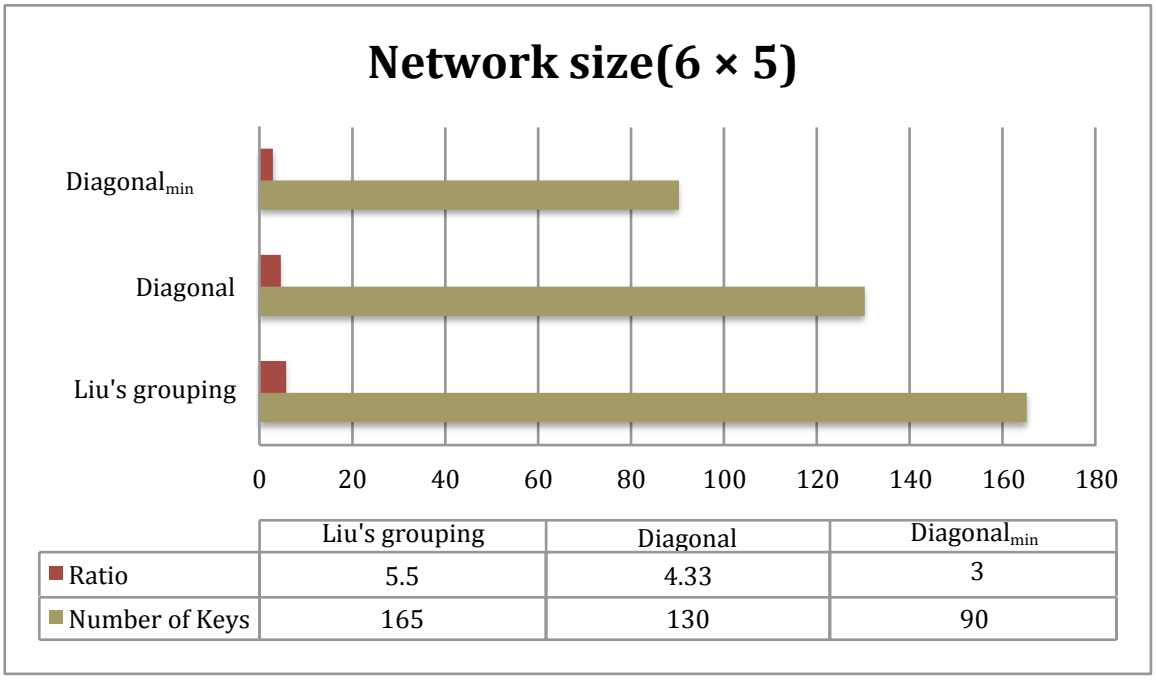
From Figure 5.17 we can see that in a 4×3 network, Liu's grouping requires 42 keys to be stored with a ratio of 3.5 keys per sensor. Diagonal-based grouping requires 32 keys to be stored with a ratio of 2.67 while diagonal$_{min}$ grouping requires 26 keys to be stored with a ratio of 2.17.



	Liu's grouping	Diagonal	Diagonal$_{min}$
Ratio	4.5	3.5	2.6
Number of Keys	90	70	52

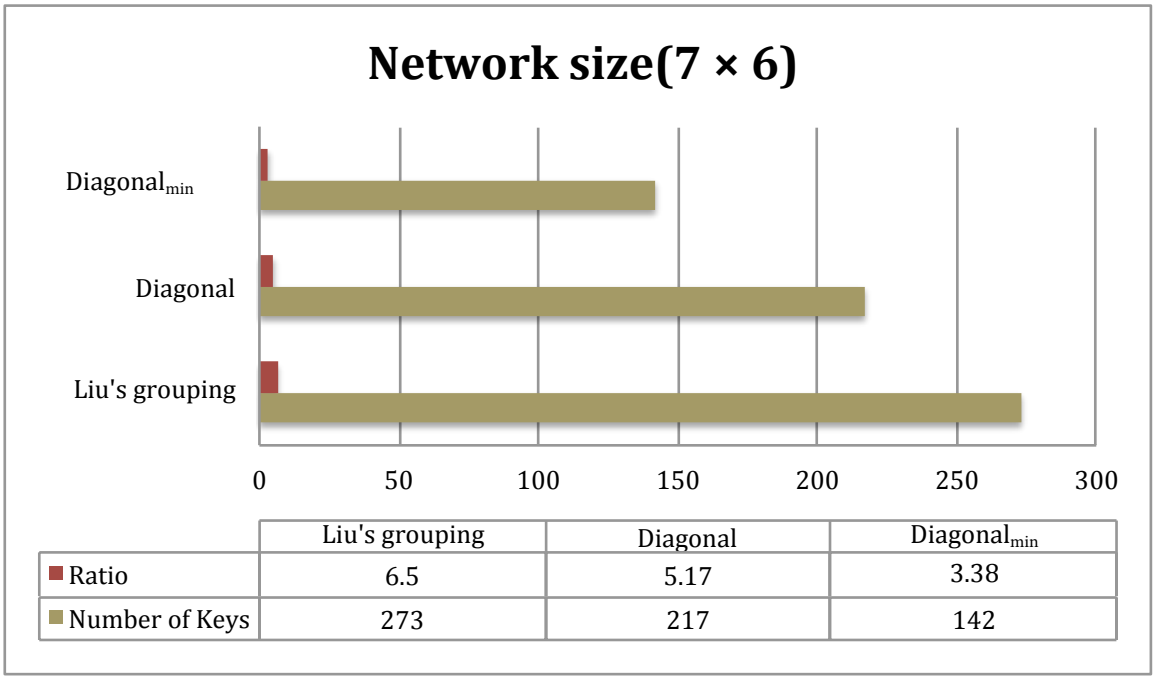
Figure 5.18: Total number of keys and key-to-sensor ratio for a 5×4 network.

For a 5×4 network, Liu's grouping requires 90 keys to be stored with a ratio of 4.5 keys per sensor. Diagonal-based grouping requires 70 keys to be stored with a ratio of 3.5 while diagonal$_{min}$ grouping requires 52 keys to be stored with a ratio of 2.6.



	Liu's grouping	Diagonal	Diagonal$_{min}$
Ratio	5.5	4.33	3
Number of Keys	165	130	90

Figure 5.19: Total number of keys and key-to-sensor ratio for a 6×5 network.



	Liu's grouping	Diagonal	Diagonal$_{min}$
Ratio	6.5	5.17	3.38
Number of Keys	273	217	142

Figure 5.20: Total number of keys and key-to-sensor ratio for a 6×7 network.

Figures 5.19 and 5.20 show that diagonal$_{min}$ keeps the key to sensor ratio around 3 even though the number of nodes has increased more than 60%.

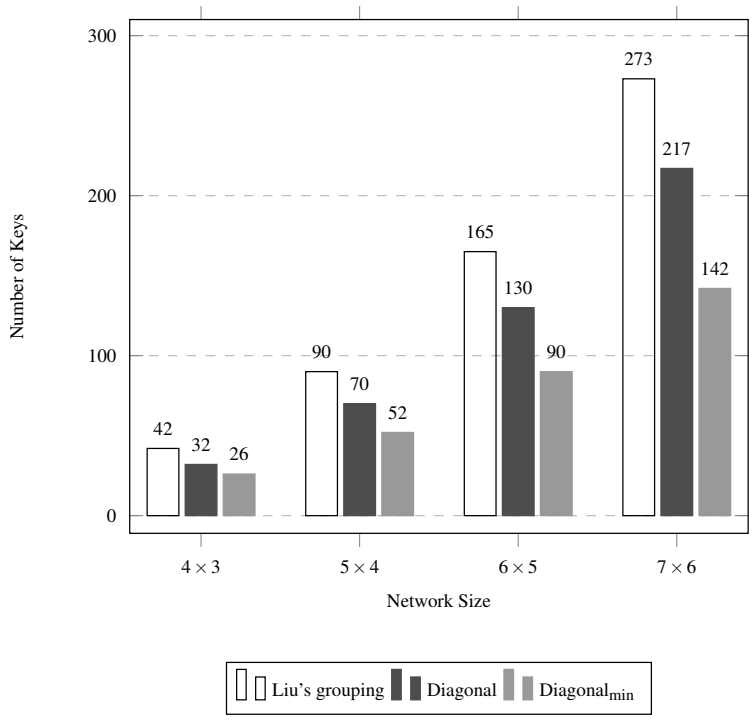


Figure 5.21: Key storage for Liu's grouping and proposed groupings where $n > m$.

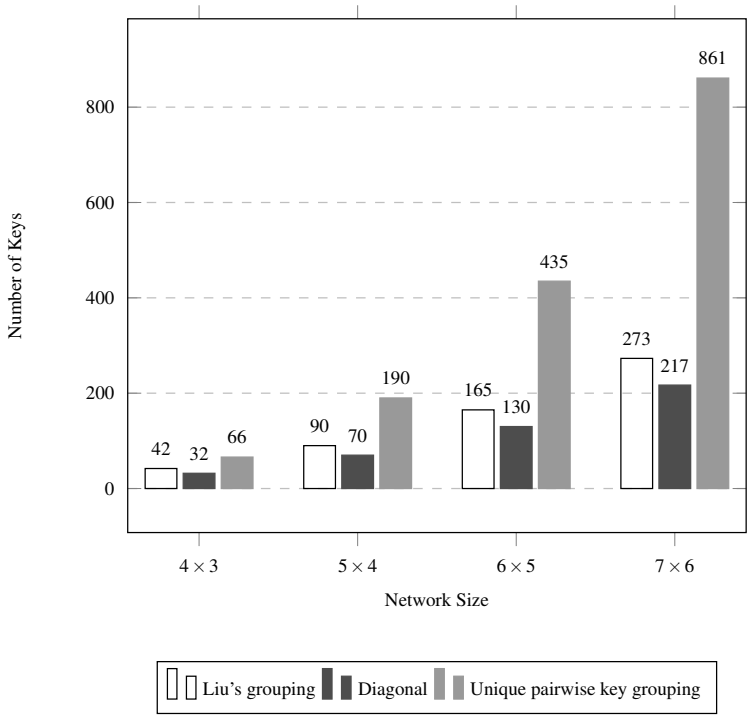


Figure 5.22: Key storage for unique pairwise key grouping and proposed groupings where $n > m$.

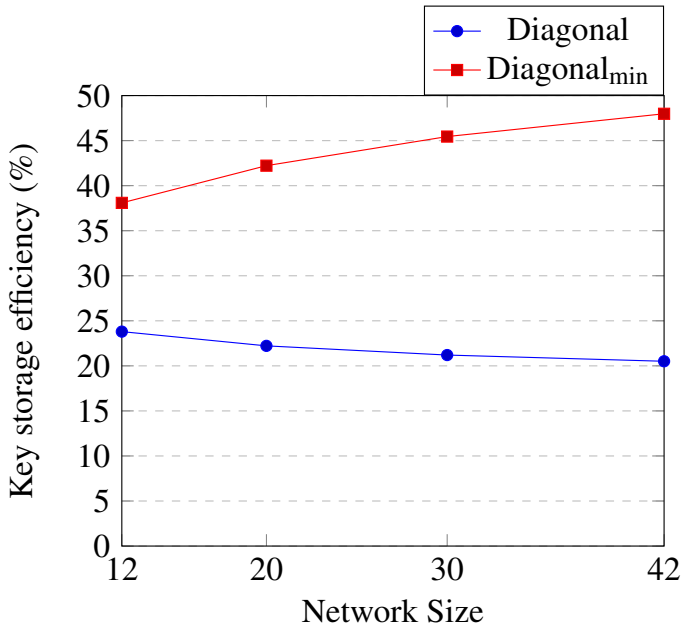


Figure 5.23: Increase in key storage efficiency where $n > m$.

If we look at the pattern of required keys for the designs in Figures 5.21, 5.22, 5.23, and 5.24 diagonal-based grouping is around 20% more key storage efficient than Liu's grouping.

Again, the required keys for $\text{diagonal}_{\text{min}}$ continues to increase as the network size grows.

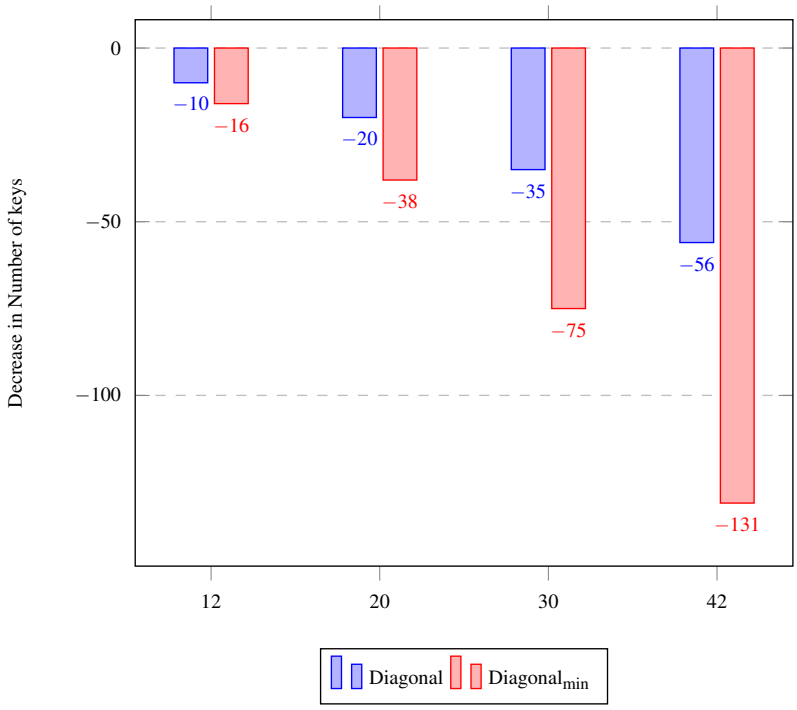


Figure 5.24: Decrease in number of keys in terms of Liu's grouping where $n > m$.

Figures 5.15 and 5.23 offer us an interesting observation. Although the total number of sensor nodes are the same for $n > m$ and $n < m$ used in our implementation, the results show that the proposed diagonal-based grouping performs better when the network has more rows than columns, that is, where $n > m$.

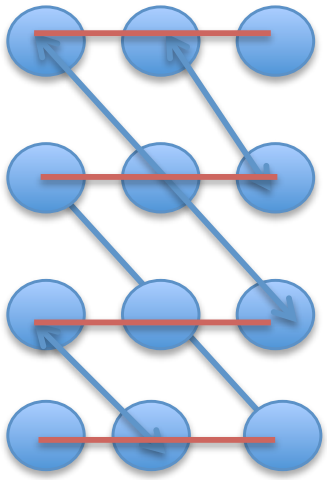


Figure 5.25: A 4×3 network.

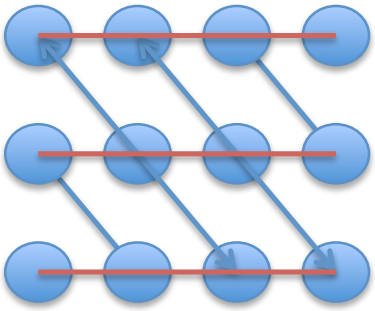


Figure 5.26: A 3×4 network.

If we look at a 4×3 network (Figure 5.25) we can see that the number of diagonals is 4 and the number of rows is also 4, while a 3×4 network (Figure 5.26) has the same number of diagonals but only 3 rows. This demonstrates that the number of rows (n) may effect the key storage of a sensor network more than the number of sensors in each row (m).

Finally, we examine the number of keys required for different network sizes as shown in Figure 5.27. Here we consider unique pairwise key grouping discussed in Chapter 3 along with Liu's grouping. It appears that in Figures 5.27 and 5.28 that as the number of nodes increases diagonal-based grouping require fewer keys than Liu's grouping and unique pairwise key grouping.

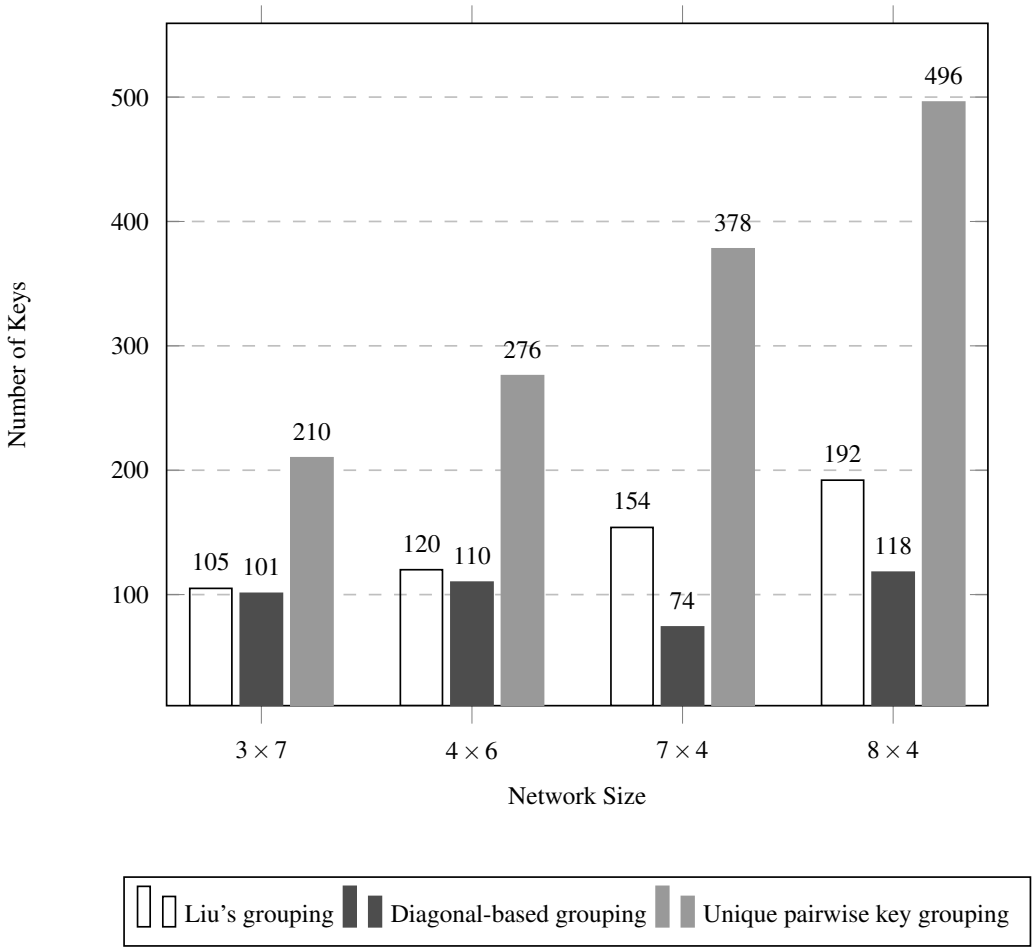


Figure 5.27: Required keys for different network sizes.

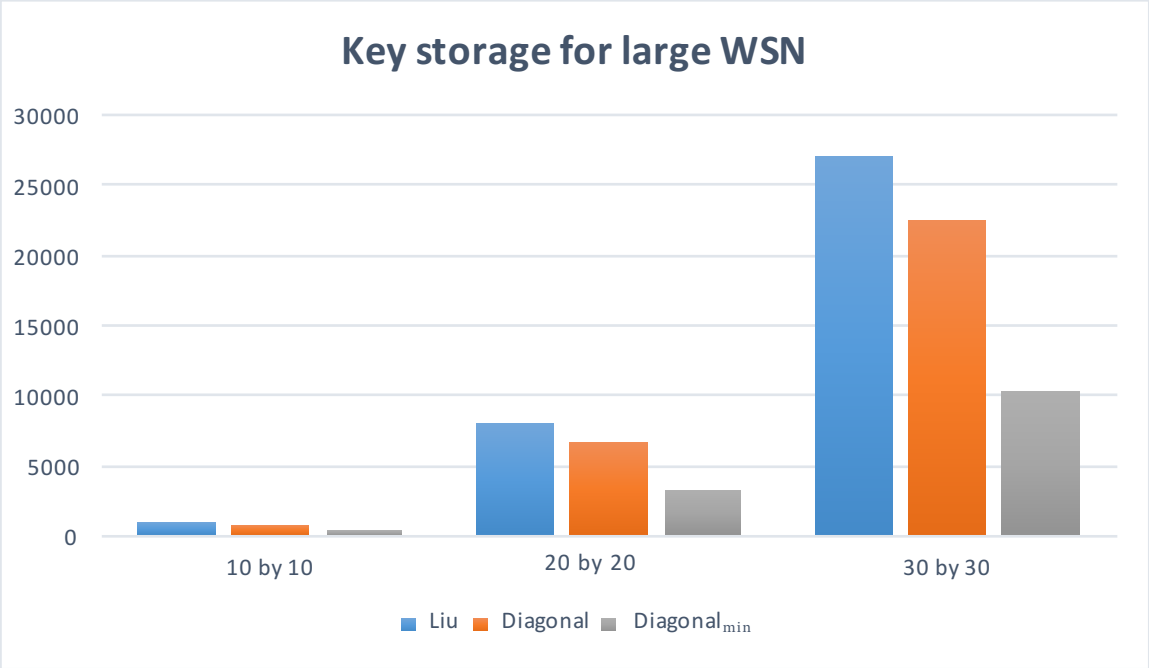


Figure 5.28: Key storage for large WSNs.

5.3 Pathkey Length

As well as the minimum storage capacity, a researcher should consider the energy scarcity of nodes which imposes the requirement of the least number of links (hops) between the sensors to reach the basestation. Although we can achieve a grouping that allows sensors to store a minimum number of keys, if it forces sensors to communicate frequently then that will drain the limited power available to the sensors. This will reduce the network lifetime. Our proposed diagonal-based grouping has also considered this requirement and shows promising results.

When two nodes are trying to communicate with each other and do not share a common key, they use a path where each pair of nodes on the path shares a key. The length of this path is called *pathkey length* [52]. Average pathkey length is important in terms of performance and network life span.

To test the required pathkey we set up a 3×3 network. All the sensors send data to every sensor one by one. That is, all sensors will send data to sensor 1, and then the destination for all the sensors is sensor 2, and so on. We counted the number of hops for each sensor and the number of keys required to reach the destination. These rules and the path map are provided in the Tables 5.1 to 5.9.

Table 5.1: Number of hops for a 3×3 network using Liu's grouping (**Destination Sensor: 1**).

Node ID	Number of Hops	Path	Number of keys
1	0	0	0
2	1	$2 \rightarrow 1$	1
3	2	$3 \rightarrow 2 \rightarrow 1$	1
4	1	$4 \rightarrow 1$	1
5	2	$5 \rightarrow 2 \rightarrow 1$	2
6	3	$6 \rightarrow 3 \rightarrow 2 \rightarrow 1$	2
7	2	$7 \rightarrow 4 \rightarrow 1$	1
8	3	$8 \rightarrow 5 \rightarrow 2 \rightarrow 1$	2
9	4	$9 \rightarrow 6 \rightarrow 3 \rightarrow 2 \rightarrow 1$	2
Total Hops	18		12

Table 5.2: Number of hops for a 3×3 network using Liu's grouping (**Destination Sensor: 2**).

Node ID	Number of Hops	Path	Number of keys
1	1	$1 \rightarrow 2$	1
2	0	0	0
3	1	$3 \rightarrow 2$	1
4	2	$4 \rightarrow 1 \rightarrow 2$	2
5	1	$5 \rightarrow 2$	1
6	2	$6 \rightarrow 3 \rightarrow 2$	2
7	3	$7 \rightarrow 4 \rightarrow 1 \rightarrow 2$	2
8	2	$8 \rightarrow 5 \rightarrow 2$	1
9	3	$9 \rightarrow 6 \rightarrow 3 \rightarrow 2$	2
Total Hops	15		12

Table 5.3: Number of hops for a 3×3 network using Liu's grouping (**Destination Sensor: 3**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 2 \rightarrow 3$	1
2	1	$2 \rightarrow 1$	1
3	0	0	0
4	3	$4 \rightarrow 1 \rightarrow 2 \rightarrow 3$	2
5	2	$5 \rightarrow 2 \rightarrow 3$	2
6	1	$6 \rightarrow 3$	1
7	4	$7 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 3$	2
8	3	$8 \rightarrow 5 \rightarrow 2 \rightarrow 3$	2
9	2	$9 \rightarrow 6 \rightarrow 3$	1
Total Hops	18		12

Table 5.4: Number of hops for a 3×3 network using Liu's grouping (**Destination Sensor: 4**).

Node ID	Number of Hops	Path	Number of keys
1	1	$1 \rightarrow 4$	1
2	2	$2 \rightarrow 1 \rightarrow 4$	2
3	3	$3 \rightarrow 2 \rightarrow 1 \rightarrow 4$	2
4	0	0	0
5	1	$5 \rightarrow 4$	1
6	2	$6 \rightarrow 5 \rightarrow 4$	1
7	1	$7 \rightarrow 4$	1
8	2	$8 \rightarrow 7 \rightarrow 4$	2
9	3	$9 \rightarrow 6 \rightarrow 5 \rightarrow 4$	2
Total Hops	15		12

Table 5.5: Number of hops for a 3×3 network using Liu's grouping (**Destination Sensor: 5**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 2 \rightarrow 5$	2
2	1	$2 \rightarrow 5$	1
3	2	$3 \rightarrow 2 \rightarrow 5$	2
4	1	$4 \rightarrow 5$	1
5	0	0	0
6	1	$6 \rightarrow 5$	1
7	2	$7 \rightarrow 4 \rightarrow 5$	2
8	1	$8 \rightarrow 5$	1
9	2	$9 \rightarrow 6 \rightarrow 5$	2
Total Hops	12		12

Table 5.6: Number of hops for a 3×3 network using Liu's grouping (**Destination Sensor: 6**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 2 \rightarrow 3 \rightarrow 6$	2
2	2	$2 \rightarrow 3 \rightarrow 6$	2
3	1	$3 \rightarrow 6$	1
4	2	$4 \rightarrow 5 \rightarrow 6$	1
5	1	$5 \rightarrow 6$	1
6	0	0	0
7	3	$7 \rightarrow 4 \rightarrow 5 \rightarrow 6$	2
8	2	$8 \rightarrow 5 \rightarrow 6$	2
9	1	$9 \rightarrow 6$	1
Total Hops	15		12

Table 5.7: Number of hops for a 3×3 network using Liu's grouping (**Destination Sensor: 7**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 4 \rightarrow 7$	1
2	3	$2 \rightarrow 1 \rightarrow 4 \rightarrow 7$	2
3	4	$3 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 7$	2
4	1	$4 \rightarrow 7$	1
5	2	$5 \rightarrow 4 \rightarrow 7$	2
6	3	$6 \rightarrow 5 \rightarrow 4 \rightarrow 7$	2
7	0	0	0
8	1	$8 \rightarrow 7$	1
9	2	$9 \rightarrow 8 \rightarrow 7$	1
Total Hops	18		12

Table 5.8: Number of hops for a 3×3 network using Liu's grouping (**Destination Sensor: 8**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 2 \rightarrow 5 \rightarrow 8$	2
2	2	$2 \rightarrow 5 \rightarrow 8$	1
3	3	$3 \rightarrow 2 \rightarrow 5 \rightarrow 8$	2
4	2	$4 \rightarrow 5 \rightarrow 8$	2
5	1	$5 \rightarrow 8$	1
6	2	$6 \rightarrow 5 \rightarrow 8$	2
7	1	$7 \rightarrow 8$	1
8	0	0	0
9	2	$9 \rightarrow 8 \rightarrow 7$	1
Total Hops	16		12

Table 5.9: Number of hops for a 3×3 network using Liu's grouping (**Destination Sensor: 9**).

Node ID	Number of Hops	Path	Number of keys
1	4	$1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 9$	2
2	3	$2 \rightarrow 3 \rightarrow 6 \rightarrow 9$	2
3	2	$3 \rightarrow 6 \rightarrow 9$	1
4	3	$4 \rightarrow 5 \rightarrow 6 \rightarrow 9$	2
5	2	$5 \rightarrow 6 \rightarrow 9$	2
6	1	$6 \rightarrow 9$	1
7	2	$7 \rightarrow 8 \rightarrow 9$	1
8	3	$8 \rightarrow 9$	1
9	0	0	0
Total Hops	20		12

We can calculate the average pathkey length from the tables above for a 3×3 network using Liu's grouping. This value turns out to be 16.33 as shown in Figure 5.29.

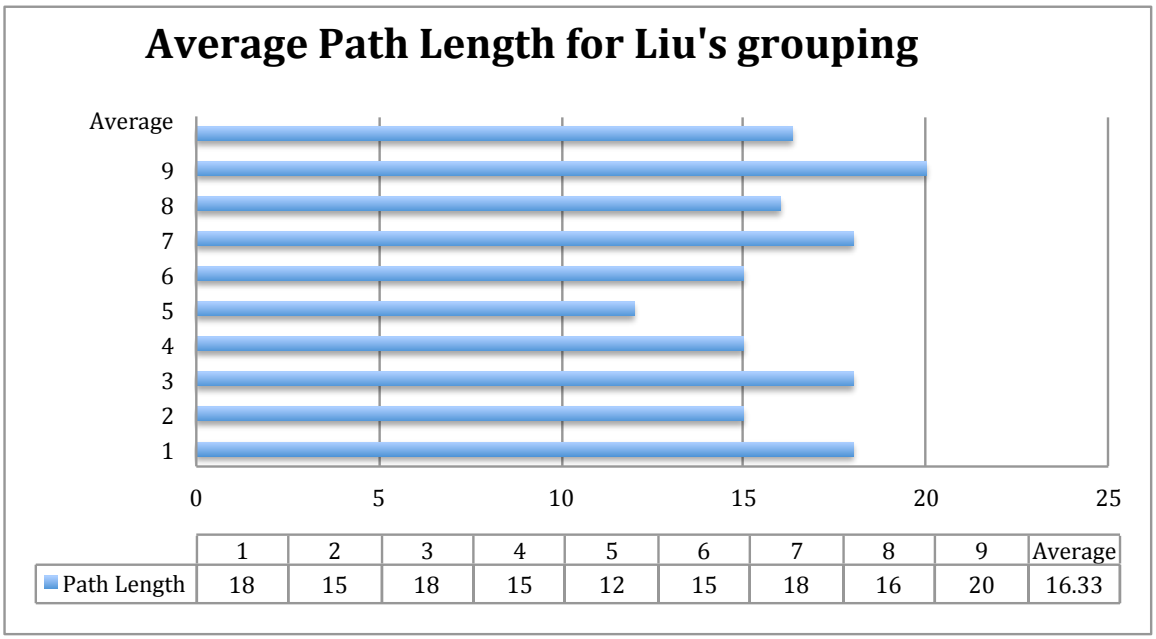


Figure 5.29: Average pathkey length using Liu's grouping for a 3×3 network, where $n = m$.

We next calculate the number of hops for a 3×3 network using diagonal-based grouping as shown in Figure 5.30.

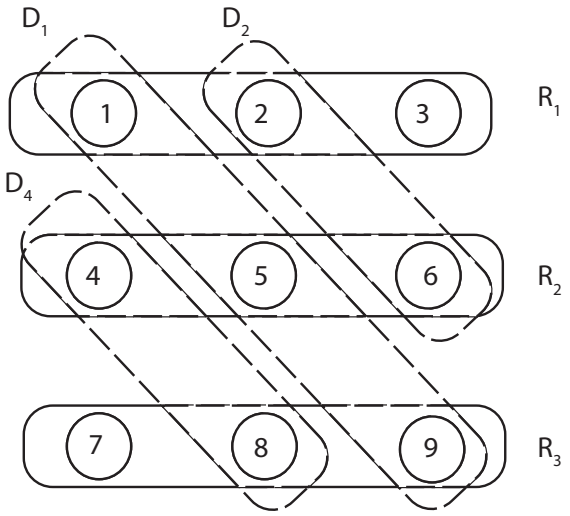


Figure 5.30: Diagonal-based grouping.

The graph tree helps us to select the path with minimum keys to encrypt as shown in Figure 5.31. The number of keys will only increase if there is a shift from the current branch to a different row or diagonal. For example path $\{1,5,9\}$ uses the same diagonal D_1 but if we shift to sensor 8, then there is a change from diagonal (D_1) to row (R_3). So the number of keys used will be 2 for path$\{1,5,9,8\}$

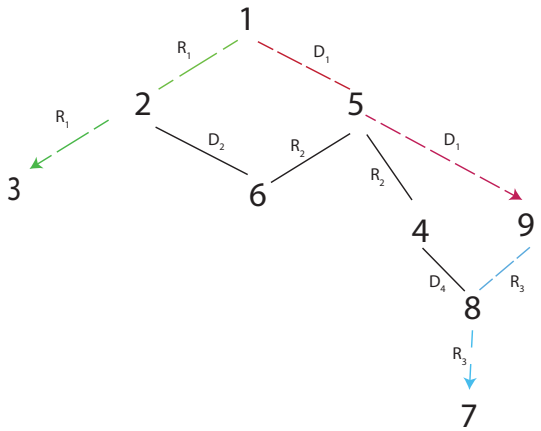


Figure 5.31: Graph tree of a diagonal-based grouping.

Table 5.10: Number of hops for a 3×3 network using diagonal-based grouping (**Destination Sensor: 1**).

Node ID	Number of Hops	Path	Number of keys
1	0	0	0
2	1	$2 \rightarrow 1$	1
3	2	$3 \rightarrow 2 \rightarrow 1$	1
4	2	$4 \rightarrow 5 \rightarrow 1$	2
5	1	$5 \rightarrow 1$	1
6	2	$6 \rightarrow 2 \rightarrow 1$	2
7	4	$7 \rightarrow 8 \rightarrow 9 \rightarrow 5 \rightarrow 1$	2
8	3	$8 \rightarrow 9 \rightarrow 5 \rightarrow 1$	2
9	2	$9 \rightarrow 5 \rightarrow 1$	1
Total Hops	17		12

Table 5.11: Number of hops for a 3×3 network using diagonal-based grouping (**Destination Sensor: 2**).

Node ID	Number of Hops	Path	Number of keys
1	1	$1 \rightarrow 2$	1
2	0	0	0
3	1	$3 \rightarrow 2$	1
4	3	$4 \rightarrow 5 \rightarrow 6 \rightarrow 2$	2
5	2	$5 \rightarrow 1 \rightarrow 2$	2
6	1	$6 \rightarrow 2$	1
7	5	$7 \rightarrow 8 \rightarrow 9 \rightarrow 5 \rightarrow 1 \rightarrow 2$	3
8	4	$8 \rightarrow 9 \rightarrow 5 \rightarrow 1 \rightarrow 2$	3
9	2	$9 \rightarrow 5 \rightarrow 1$	2
Total Hops	19		15

Table 5.12: Number of hops for a 3×3 network using diagonal-based grouping (**Destination Sensor: 3**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 2 \rightarrow 3$	1
2	1	$2 \rightarrow 3$	1
3	0	0	0
4	4	$4 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3$	3
5	3	$5 \rightarrow 1 \rightarrow 2 \rightarrow 3$	2
6	2	$6 \rightarrow 2 \rightarrow 3$	2
7	6	$7 \rightarrow 8 \rightarrow 9 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3$	3
8	5	$8 \rightarrow 9 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3$	3
9	4	$9 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3$	2
Total Hops	27		17

Table 5.13: Number of hops for a 3×3 network using diagonal-based grouping (**Destination Sensor: 4**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 5 \rightarrow 4$	2
2	3	$2 \rightarrow 6 \rightarrow 5 \rightarrow 4$	2
3	4	$3 \rightarrow 2 \rightarrow 6 \rightarrow 5 \rightarrow 4$	3
4	0	0	0
5	1	$5 \rightarrow 4$	1
6	2	$6 \rightarrow 5 \rightarrow 4$	1
7	2	$7 \rightarrow 8 \rightarrow 4$	2
8	1	$8 \rightarrow 4$	1
9	2	$9 \rightarrow 5 \rightarrow 4$	2
Total Hops	17		14

Table 5.14: Number of hops for a 3×3 network using diagonal-based grouping (**Destination Sensor: 5**).

Node ID	Number of Hops	Path	Number of keys
1	1	$1 \rightarrow 5$	1
2	2	$2 \rightarrow 6 \rightarrow 5$	2
3	3	$3 \rightarrow 2 \rightarrow 1 \rightarrow 5$	2
4	1	$4 \rightarrow 5$	1
5	0	0	0
6	1	$6 \rightarrow 5$	1
7	3	$7 \rightarrow 8 \rightarrow 9 \rightarrow 5$	2
8	2	$8 \rightarrow 4 \rightarrow 5$	2
9	1	$9 \rightarrow 5$	1
Total Hops	14		12

Table 5.15: Number of hops for a 3×3 network using diagonal-based grouping (**Destination Sensor: 6**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 5 \rightarrow 6$	2
2	1	$2 \rightarrow 6$	1
3	2	$3 \rightarrow 2 \rightarrow 6$	2
4	2	$4 \rightarrow 5 \rightarrow 6$	1
5	1	$5 \rightarrow 6$	1
6	0	0	0
7	4	$7 \rightarrow 8 \rightarrow 4 \rightarrow 5 \rightarrow 6$	3
8	3	$8 \rightarrow 4 \rightarrow 5 \rightarrow 6$	2
9	2	$9 \rightarrow 5 \rightarrow 6$	2
Total Hops	17		14

Table 5.16: Number of hops for a 3×3 network using diagonal-based grouping (**Destination Sensor: 7**).

Node ID	Number of Hops	Path	Number of keys
1	4	$1 \rightarrow 5 \rightarrow 9 \rightarrow 8 \rightarrow 7$	2
2	5	$2 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 8 \rightarrow 7$	3
3	6	$3 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 8 \rightarrow 7$	3
4	2	$4 \rightarrow 8 \rightarrow 7$	2
5	3	$5 \rightarrow 9 \rightarrow 8 \rightarrow 7$	2
6	4	$6 \rightarrow 5 \rightarrow 9 \rightarrow 8 \rightarrow 7$	3
7	0	0	0
8	1	$8 \rightarrow 7$	1
9	2	$9 \rightarrow 8 \rightarrow 7$	1
Total Hops	27		17

Table 5.17: Number of hops for a 3×3 network using diagonal-based grouping (**Destination Sensor: 8**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 5 \rightarrow 9 \rightarrow 8$	2
2	4	$2 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 8$	3
3	5	$3 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 8$	3
4	1	$4 \rightarrow 8$	1
5	2	$5 \rightarrow 9 \rightarrow 8$	2
6	3	$6 \rightarrow 5 \rightarrow 4 \rightarrow 8$	2
7	1	$7 \rightarrow 8$	1
8	0	0	0
9	1	$9 \rightarrow 8$	1
Total Hops	20		15

Table 5.18: Number of hops for a 3×3 network using diagonal-based grouping (**Destination Sensor: 9**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 5 \rightarrow 9$	1
2	3	$2 \rightarrow 1 \rightarrow 5 \rightarrow 9$	2
3	4	$3 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 9$	2
4	2	$4 \rightarrow 8 \rightarrow 9$	2
5	1	$5 \rightarrow 9$	1
6	2	$6 \rightarrow 5 \rightarrow 9$	2
7	2	$7 \rightarrow 8 \rightarrow 9$	1
8	1	$8 \rightarrow 9$	1
9	0	0	0
Total Hops	17		12

We can calculate the average pathkey length from Tables 5.10 to 5.18 for a 3×3 network using diagonal-based grouping. This value turns out to be 19.44, as shown in Figure 5.32.

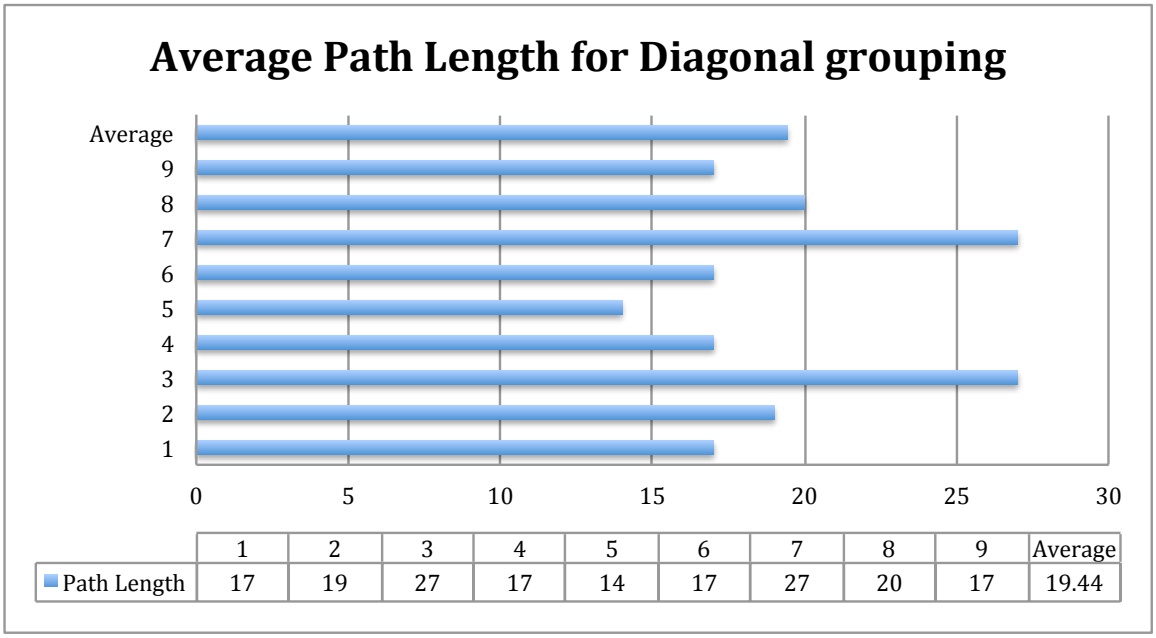


Figure 5.32: Average pathkey length using diagonal-based grouping for a 3×3 network, where $n = m$.

We next calculate the number of hops for a 3×3 network using diagonal$_{\text{min}}$ grouping

as shown in Figure 5.33.

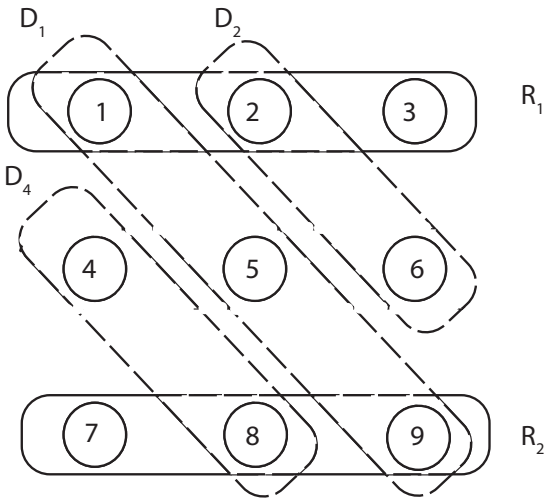


Figure 5.33: Diagonal$_{\text{min}}$ grouping.

Table 5.19: Number of hops for a 3×3 network using diagonal$_{\text{min}}$ grouping (**Destination Sensor: 1**).

Node ID	Number of Hops	Path	Number of keys
1	0	0	0
2	1	$2 \rightarrow 1$	1
3	2	$3 \rightarrow 2 \rightarrow 1$	1
4	4	$4 \rightarrow 8 \rightarrow 9 \rightarrow 5 \rightarrow 1$	3
5	1	$5 \rightarrow 1$	1
6	2	$6 \rightarrow 2 \rightarrow 1$	2
7	4	$7 \rightarrow 8 \rightarrow 9 \rightarrow 5 \rightarrow 1$	2
8	3	$8 \rightarrow 9 \rightarrow 5 \rightarrow 1$	2
9	2	$9 \rightarrow 5 \rightarrow 1$	1
Total Hops	19		13

Table 5.20: Number of hops for a 3×3 network using $\text{diagonal}_{\text{min}}$ grouping (**Destination Sensor: 2**).

Node ID	Number of Hops	Path	Number of keys
1	1	$1 \rightarrow 2$	1
2	0	0	0
3	2	$3 \rightarrow 2$	1
4	5	$4 \rightarrow 8 \rightarrow 9 \rightarrow 5 \rightarrow 1 \rightarrow 2$	4
5	2	$5 \rightarrow 1 \rightarrow 2$	2
6	1	$6 \rightarrow 2$	1
7	5	$7 \rightarrow 8 \rightarrow 9 \rightarrow 5 \rightarrow 1 \rightarrow 2$	3
8	4	$8 \rightarrow 9 \rightarrow 5 \rightarrow 1 \rightarrow 2$	3
9	3	$9 \rightarrow 5 \rightarrow 1 \rightarrow 2$	2
Total Hops	23		17

Table 5.21: Number of hops for a 3×3 network using $\text{diagonal}_{\text{min}}$ grouping (**Destination Sensor: 3**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 2 \rightarrow 3$	1
2	1	$2 \rightarrow 3$	1
3	0	0	0
4	6	$4 \rightarrow 8 \rightarrow 9 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3$	4
5	2	$5 \rightarrow 1 \rightarrow 2 \rightarrow 3$	2
6	2	$6 \rightarrow 2 \rightarrow 3$	2
7	6	$7 \rightarrow 8 \rightarrow 9 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3$	3
8	5	$8 \rightarrow 9 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3$	3
9	4	$9 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3$	2
Total Hops	28		18

Table 5.22: Number of hops for a 3×3 network using diagonal$_{\text{min}}$ grouping (**Destination Sensor: 4**).

Node ID	Number of Hops	Path	Number of keys
1	4	$1 \rightarrow 5 \rightarrow 9 \rightarrow 8 \rightarrow 4$	3
2	5	$2 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 8 \rightarrow 4$	3
3	6	$3 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 8 \rightarrow 4$	3
4	0	0	0
5	3	$5 \rightarrow 9 \rightarrow 8 \rightarrow 4$	3
6	6	$6 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 8 \rightarrow 4$	5
7	2	$7 \rightarrow 8 \rightarrow 4$	2
8	1	$8 \rightarrow 4$	1
9	2	$9 \rightarrow 8 \rightarrow 4$	2
Total Hops	29		22

Table 5.23: Number of hops for a 3×3 network using diagonal$_{\text{min}}$ grouping (**Destination Sensor: 5**).

Node ID	Number of Hops	Path	Number of keys
1	1	$1 \rightarrow 5$	1
2	2	$2 \rightarrow 1 \rightarrow 5$	2
3	3	$3 \rightarrow 2 \rightarrow 1 \rightarrow 5$	2
4	3	$4 \rightarrow 8 \rightarrow 9 \rightarrow 5$	3
5	0	0	0
6	3	$6 \rightarrow 2 \rightarrow 1 \rightarrow 5$	3
7	3	$7 \rightarrow 8 \rightarrow 9 \rightarrow 5$	2
8	2	$8 \rightarrow 9 \rightarrow 5$	2
9	1	$9 \rightarrow 5$	1
Total Hops	18		16

Table 5.24: Number of hops for a 3×3 network using $\text{diagonal}_{\text{min}}$ grouping (**Destination Sensor: 6**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 2 \rightarrow 6$	2
2	1	$2 \rightarrow 6$	1
3	2	$3 \rightarrow 2 \rightarrow 6$	2
4	6	$4 \rightarrow 8 \rightarrow 9 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 6$	5
5	3	$5 \rightarrow 1 \rightarrow 2 \rightarrow 6$	3
6	0	0	0
7	6	$7 \rightarrow 8 \rightarrow 9 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 6$	4
8	5	$8 \rightarrow 9 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 6$	4
9	4	$9 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 6$	3
Total Hops	29		24

Table 5.25: Number of hops for a 3×3 network using $\text{diagonal}_{\text{min}}$ grouping (**Destination Sensor: 7**).

Node ID	Number of Hops	Path	Number of keys
1	4	$1 \rightarrow 5 \rightarrow 9 \rightarrow 8 \rightarrow 7$	2
2	5	$2 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 8 \rightarrow 7$	3
3	6	$3 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 8 \rightarrow 7$	3
4	1	$4 \rightarrow 7$	2
5	3	$5 \rightarrow 9 \rightarrow 8 \rightarrow 7$	2
6	6	$6 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 8 \rightarrow 7$	5
7	0	0	0
8	1	$8 \rightarrow 7$	1
9	2	$9 \rightarrow 8 \rightarrow 7$	1
Total Hops	27		19

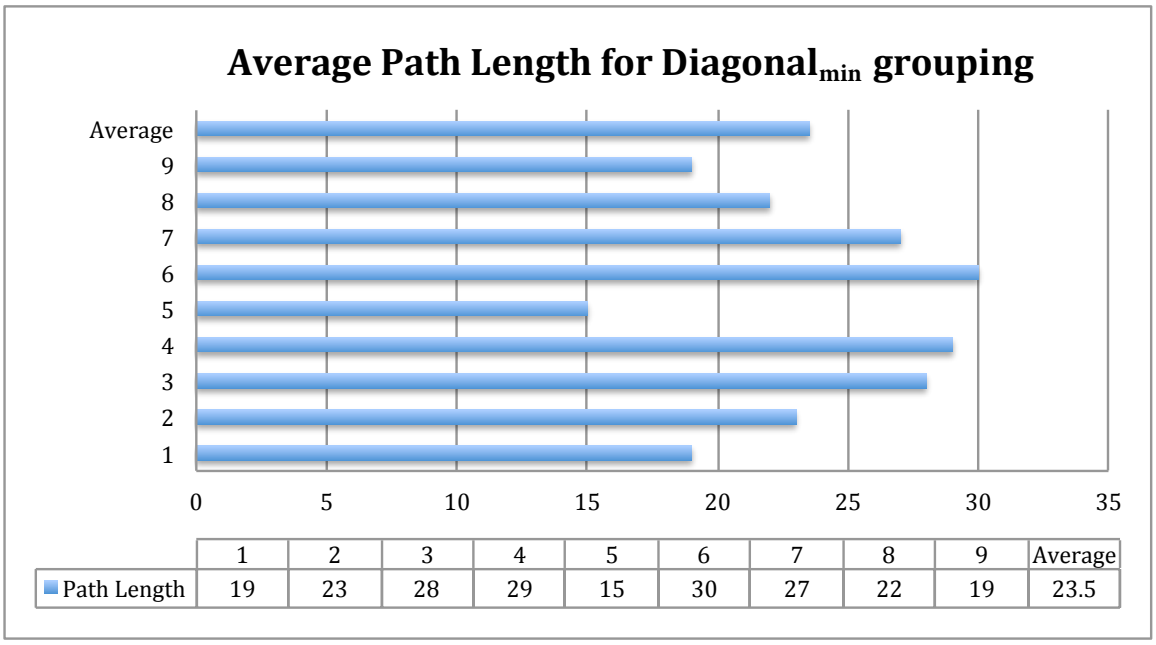
Table 5.26: Number of hops for a 3×3 network using $\text{diagonal}_{\text{min}}$ grouping (**Destination Sensor: 8**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 5 \rightarrow 9 \rightarrow 8$	2
2	4	$2 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 8$	3
3	5	$3 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 8$	3
4	1	$4 \rightarrow 8$	1
5	2	$5 \rightarrow 9 \rightarrow 8$	2
6	5	$3 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 9 \rightarrow 8$	4
7	1	$7 \rightarrow 8$	1
8	0	0	0
9	1	$9 \rightarrow 8$	1
Total Hops	22		17

Table 5.27: Number of hops for a 3×3 network using $\text{diagonal}_{\text{min}}$ grouping (**Destination Sensor: 9**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 5 \rightarrow 9$	1
2	3	$2 \rightarrow 1 \rightarrow 5 \rightarrow 9$	2
3	4	$3 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 9$	2
4	2	$4 \rightarrow 8 \rightarrow 9$	2
5	1	$5 \rightarrow 9$	1
6	4	$6 \rightarrow 2 \rightarrow 1 \rightarrow 5 \rightarrow 9$	3
7	2	$7 \rightarrow 8 \rightarrow 9$	1
8	1	$8 \rightarrow 9$	1
9	0	0	0
Total Hops	19		13

We can calculate the average pathkey length from the Tables 5.19 to 5.27 for a 3×3 network using diagonal$_{\text{min}}$ grouping. This value turns out to be 23.5, as shown in Figure 5.34.



	1	2	3	4	5	6	7	8	9	Average
Path Length	19	23	28	29	15	30	27	22	19	23.5

Figure 5.34: Average pathkey length using diagonal$_{\text{min}}$ grouping for a 3×3 network, where $n = m$.

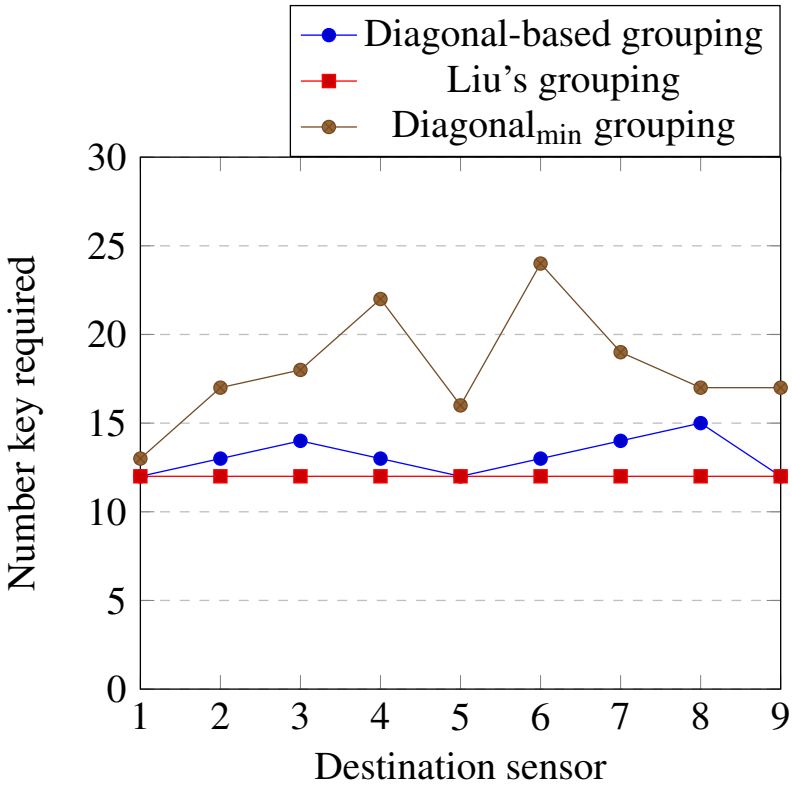


Figure 5.35: Comparison of number of keys utilized.

From Figures 5.35 and 5.36 we can see that the proposed grouping performs close to Liu's grouping when there are lots of transmissions along the diagonal direction. Sensor 3 and 7 do not belong to a diagonal. That is why their required keys are high. For diagonal$_{min}$, sensor 4 and 6 do not belong to any rows resulting in high key requirement.

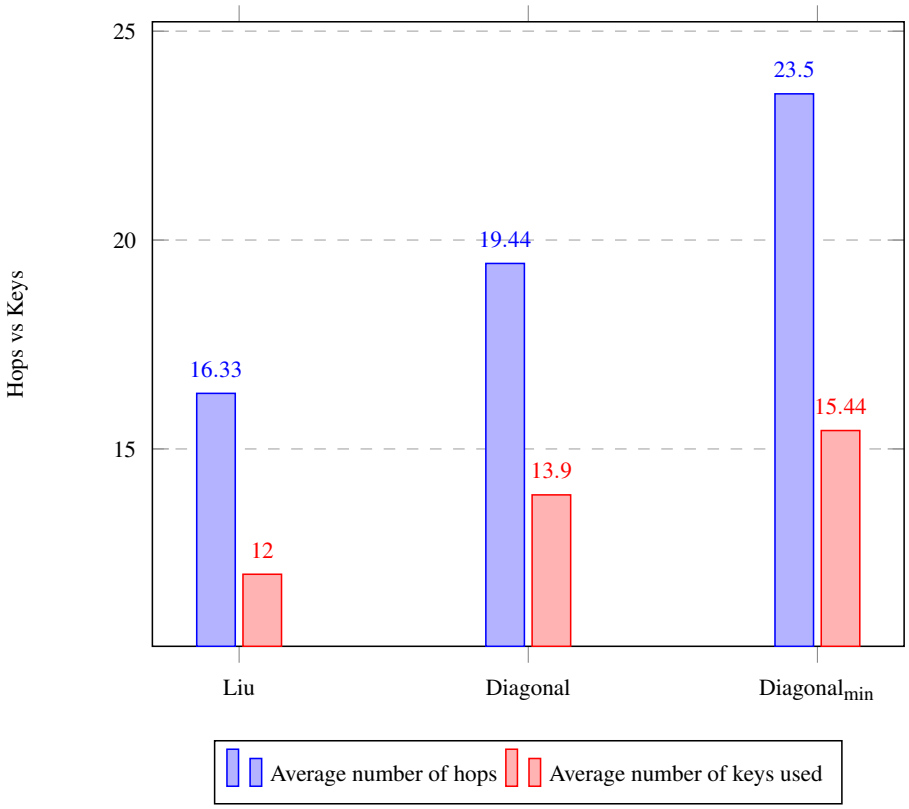


Figure 5.36: Comparison between average number of hops and average number of keys used 3×3 network, where $n = m$.

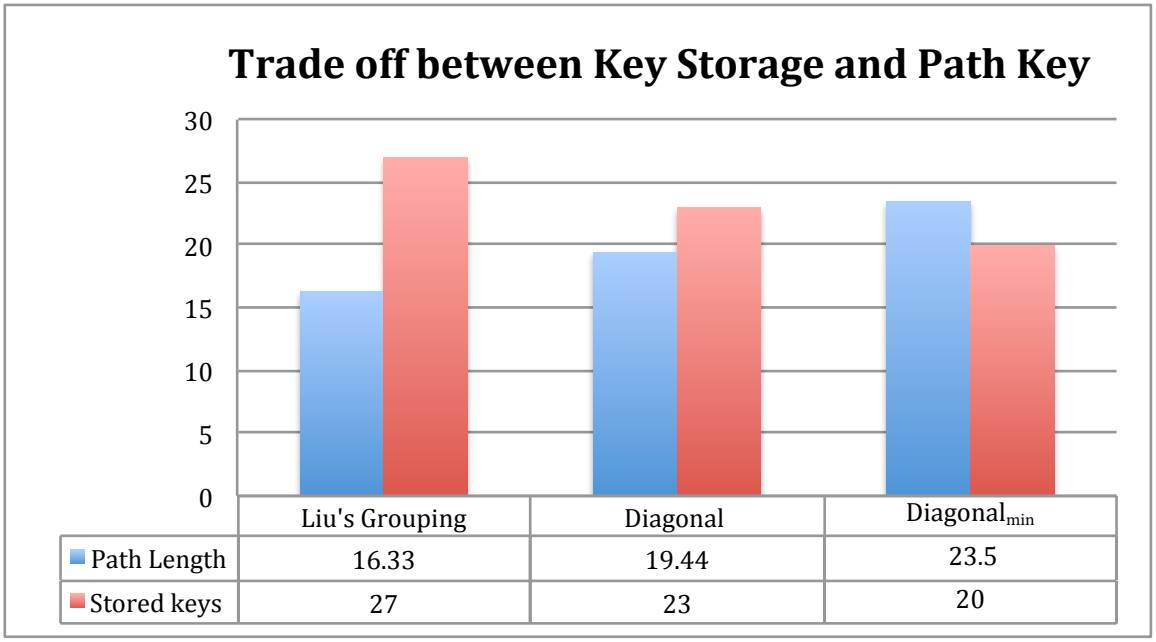


Figure 5.37: Trade-off between different groupings for 3×3 network, where $n = m$.

We next calculate the number of hops for a 3×4 network, where $n < m$ using Liu's grouping, diagonal-based grouping, and diagonal$_{\text{min}}$ grouping respectively. Figure 5.38 illustrates an example of diagonal-based grouping implementation.

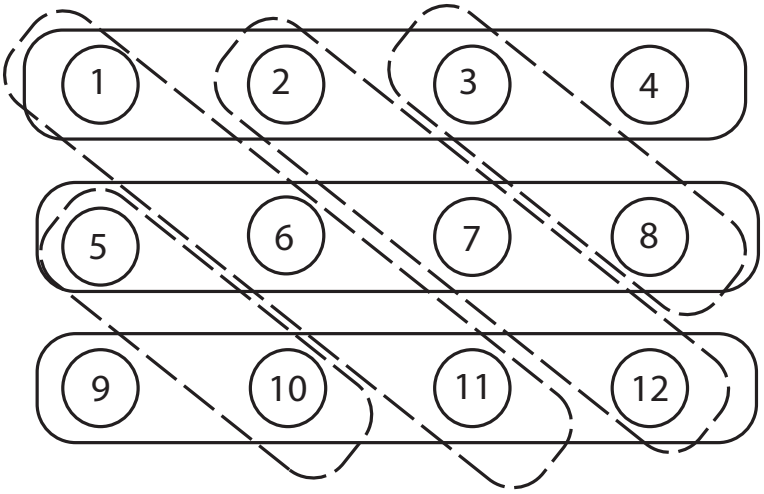


Figure 5.38: A 3×4 network, where $n < m$ using diagonal-based grouping

The graph tree as shown in Figure 5.39 helps to select the path with minimum keys to encrypt the data. The number of keys will only increase if there is a shift from the current branch to a different row or diagonal. For example Path $\{1,2,3\}$ uses the same row R_1 but if we shift to sensor 8, then there is a change from row (R_1) to diagonal (D_3). So the number of key used will be 2 for path$\{1,2,3,8\}$.

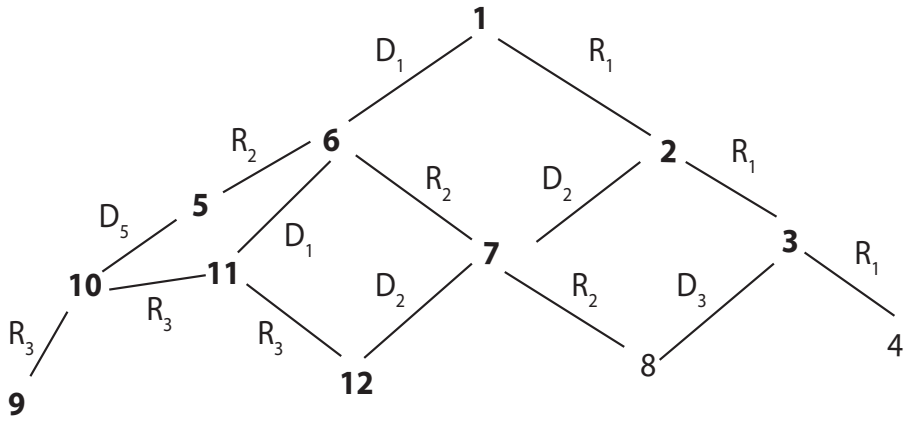


Figure 5.39: Graph tree of diagonal-based grouping.

Table 5.28: Number of hops for a 3×4 network using diagonal-based grouping (**Destination Sensor: 1**).

Node ID	Number of Hops	Path	Number of keys
1	0	0	0
2	1	$2 \rightarrow 1$	1
3	2	$3 \rightarrow 2 \rightarrow 1$	1
4	3	$4 \rightarrow 3 \rightarrow 2 \rightarrow 1$	1
5	2	$5 \rightarrow 6 \rightarrow 1$	2
6	1	$6 \rightarrow 1$	1
7	2	$7 \rightarrow 6 \rightarrow 1$	2
8	3	$8 \rightarrow 7 \rightarrow 6 \rightarrow 1$	2
9	4	$9 \rightarrow 10 \rightarrow 11 \rightarrow 6 \rightarrow 1$	2
10	3	$10 \rightarrow 11 \rightarrow 6 \rightarrow 1$	2
11	2	$11 \rightarrow 6 \rightarrow 1$	1
12	3	$12 \rightarrow 11 \rightarrow 6 \rightarrow 1$	2
Total Hops	26		17

Table 5.29: Number of hops for a 3×4 network using diagonal-based grouping (**Destination Sensor: 2**).

Node ID	Number of Hops	Path	Number of keys
1	1	$1 \rightarrow 2$	1
2	0	0	0
3	1	$3 \rightarrow 2$	1
4	2	$4 \rightarrow 3 \rightarrow 2$	1
5	3	$5 \rightarrow 6 \rightarrow 7 \rightarrow 2$	2
6	1	$6 \rightarrow 7 \rightarrow 2$	2
7	1	$7 \rightarrow 2$	1
8	2	$8 \rightarrow 7 \rightarrow 2$	2
9	5	$9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 7 \rightarrow 2$	2
10	4	$10 \rightarrow 11 \rightarrow 12 \rightarrow 7 \rightarrow 2$	2
11	3	$11 \rightarrow 12 \rightarrow 7 \rightarrow 2$	2
12	2	$12 \rightarrow 7 \rightarrow 2$	1
Total Hops	25		17

Table 5.30: Number of hops for a 3×4 network using diagonal-based grouping (**Destination Sensor: 3**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 2 \rightarrow 3$	1
2	1	$2 \rightarrow 3$	1
3	0	0	0
4	1	$4 \rightarrow 3$	1
5	4	$5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 3$	2
6	3	$6 \rightarrow 7 \rightarrow 8 \rightarrow 3$	2
7	2	$7 \rightarrow 8 \rightarrow 3$	2
8	1	$8 \rightarrow 3$	1
9	6	$9 \rightarrow 10 \rightarrow 11 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3$	3
10	5	$10 \rightarrow 11 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3$	3
11	4	$11 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3$	2
12	3	$12 \rightarrow 7 \rightarrow 2 \rightarrow 3$	2
Total Hops	32		20

Table 5.31: Number of hops for a 3×4 network using diagonal-based grouping (**Destination Sensor: 4**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	1
2	2	$2 \rightarrow 3 \rightarrow 4$	1
3	1	$3 \rightarrow 4$	1
4	0	0	0
5	5	$5 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	3
6	4	$6 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	2
7	3	$7 \rightarrow 2 \rightarrow 3 \rightarrow 4$	2
8	2	$8 \rightarrow 3 \rightarrow 4$	2
9	7	$9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 7 \rightarrow 2 \rightarrow 3 \rightarrow 4$	3
10	6	$10 \rightarrow 5 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	4
11	5	$11 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	2
12	4	$12 \rightarrow 7 \rightarrow 2 \rightarrow 3 \rightarrow 4$	2
Total Hops	42		24

Table 5.32: Number of hops for a 3×4 network using diagonal-based grouping (**Destination Sensor: 5**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 6 \rightarrow 5$	2
2	3	$2 \rightarrow 7 \rightarrow 6 \rightarrow 5$	2
3	4	$3 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 5$	2
4	5	$4 \rightarrow 3 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 5$	3
5	0	0	0
6	1	$6 \rightarrow 5$	1
7	2	$7 \rightarrow 6 \rightarrow 5$	1
8	3	$8 \rightarrow 7 \rightarrow 6 \rightarrow 5$	1
9	2	$9 \rightarrow 10 \rightarrow 5$	2
10	1	$10 \rightarrow 5$	1
11	2	$11 \rightarrow 10 \rightarrow 5$	2
12	3	$12 \rightarrow 7 \rightarrow 6 \rightarrow 5$	2
Total Hops	28		19

Table 5.33: Number of hops for a 3×4 network using diagonal-based grouping (**Destination Sensor: 6**).

Node ID	Number of Hops	Path	Number of keys
1	1	$1 \rightarrow 6$	1
2	2	$2 \rightarrow 7 \rightarrow 6$	2
3	3	$3 \rightarrow 8 \rightarrow 7 \rightarrow 6$	2
4	4	$4 \rightarrow 3 \rightarrow 8 \rightarrow 7 \rightarrow 6$	3
5	1	$5 \rightarrow 6$	1
6	0	0	0
7	1	$7 \rightarrow 6$	1
8	2	$8 \rightarrow 7 \rightarrow 6$	1
9	3	$9 \rightarrow 10 \rightarrow 11 \rightarrow 6$	2
10	2	$10 \rightarrow 11 \rightarrow 6$	2
11	1	$11 \rightarrow 6$	1
12	2	$12 \rightarrow 11 \rightarrow 6$	2
Total Hops	22		18

Table 5.34: Number of hops for a 3×4 network using diagonal-based grouping (**Destination Sensor: 7**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 6 \rightarrow 7$	2
2	1	$2 \rightarrow 7$	1
3	2	$3 \rightarrow 8 \rightarrow 7$	2
4	3	$4 \rightarrow 3 \rightarrow 8 \rightarrow 7$	3
5	2	$5 \rightarrow 6 \rightarrow 7$	1
6	1	$6 \rightarrow 7$	1
7	0	0	0
8	2	$8 \rightarrow 7$	1
9	4	$9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 7$	2
10	3	$10 \rightarrow 11 \rightarrow 12 \rightarrow 7$	2
11	2	$11 \rightarrow 12 \rightarrow 7$	2
12	1	$12 \rightarrow 7$	1
Total Hops	23		18

Table 5.35: Number of hops for a 3×4 network using diagonal-based grouping (**Destination Sensor: 8**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 6 \rightarrow 7 \rightarrow 8$	2
2	2	$2 \rightarrow 7 \rightarrow 8$	2
3	3	$3 \rightarrow 8$	1
4	2	$4 \rightarrow 3 \rightarrow 8$	2
5	3	$5 \rightarrow 6 \rightarrow 7 \rightarrow 8$	1
6	2	$6 \rightarrow 7 \rightarrow 8$	1
7	1	$7 \rightarrow 8$	1
8	0	0	0
9	5	$9 \rightarrow 10 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$	3
10	4	$10 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$	2
11	3	$11 \rightarrow 6 \rightarrow 7 \rightarrow 8$	2
12	2	$12 \rightarrow 7 \rightarrow 8$	2
Total Hops	30		19

Table 5.36: Number of hops for a 3×4 network using diagonal-based grouping (**Destination Sensor: 9**).

Node ID	Number of Hops	Path	Number of keys
1	4	$1 \rightarrow 6 \rightarrow 11 \rightarrow 10 \rightarrow 9$	2
2	5	$2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10 \rightarrow 9$	2
3	6	$3 \rightarrow 2 \rightarrow 1 \rightarrow 6 \rightarrow 11 \rightarrow 10 \rightarrow 9$	3
4	7	$4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 6 \rightarrow 11 \rightarrow 10 \rightarrow 9$	3
5	2	$5 \rightarrow 10 \rightarrow 9$	2
6	3	$6 \rightarrow 11 \rightarrow 10 \rightarrow 9$	2
7	4	$7 \rightarrow 12 \rightarrow 11 \rightarrow 10 \rightarrow 9$	2
8	5	$8 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10 \rightarrow 9$	3
9	0	0	0
10	1	$10 \rightarrow 9$	1
11	2	$11 \rightarrow 10 \rightarrow 9$	1
12	3	$12 \rightarrow 11 \rightarrow 10 \rightarrow 9$	1
Total Hops	42		22

Table 5.37: Number of hops for a 3×4 network using diagonal-based grouping (**Destination Sensor: 10**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 6 \rightarrow 11 \rightarrow 10$	2
2	4	$2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10$	2
3	5	$3 \rightarrow 2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10$	3
4	6	$4 \rightarrow 3 \rightarrow 2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10$	3
5	1	$5 \rightarrow 10$	1
6	2	$6 \rightarrow 11 \rightarrow 10$	2
7	3	$7 \rightarrow 12 \rightarrow 11 \rightarrow 10$	2
8	4	$8 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10$	3
9	1	$9 \rightarrow 10$	1
10	0	0	0
11	1	$11 \rightarrow 10$	1
12	2	$12 \rightarrow 11 \rightarrow 10$	1
Total Hops	32		21

Table 5.38: Number of hops for a 3×4 network using diagonal-based grouping (**Destination Sensor: 11**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 6 \rightarrow 11$	1
2	3	$2 \rightarrow 1 \rightarrow 6 \rightarrow 11$	2
3	4	$3 \rightarrow 2 \rightarrow 1 \rightarrow 6 \rightarrow 11$	2
4	5	$4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 6 \rightarrow 11$	2
5	2	$5 \rightarrow 6 \rightarrow 11$	2
6	1	$6 \rightarrow 11$	1
7	2	$7 \rightarrow 6 \rightarrow 11$	2
8	3	$8 \rightarrow 7 \rightarrow 6 \rightarrow 11$	2
9	2	$9 \rightarrow 10 \rightarrow 11$	1
10	1	$10 \rightarrow 11$	1
11	0	0	0
12	1	$12 \rightarrow 11$	1
Total Hops	26		17

Table 5.39: Number of hops for a 3×4 network using diagonal-based grouping (**Destination Sensor: 12**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 6 \rightarrow 11 \rightarrow 12$	2
2	2	$2 \rightarrow 7 \rightarrow 12$	1
3	3	$3 \rightarrow 2 \rightarrow 7 \rightarrow 12$	2
4	4	$4 \rightarrow 3 \rightarrow 2 \rightarrow 7 \rightarrow 12$	2
5	3	$5 \rightarrow 6 \rightarrow 7 \rightarrow 12$	2
6	2	$6 \rightarrow 7 \rightarrow 12$	2
7	1	$7 \rightarrow 12$	1
8	2	$8 \rightarrow 7 \rightarrow 12$	2
9	3	$9 \rightarrow 10 \rightarrow 11 \rightarrow 12$	1
10	2	$10 \rightarrow 11 \rightarrow 12$	1
11	1	$11 \rightarrow 12$	1
12	0	0	0
Total Hops	26		17

We can calculate the average pathkey length from Tables 5.28 to 5.39 for a 3×4 network using diagonal-based grouping. This value turns out to be 29.7, as shown in Figure 5.40.

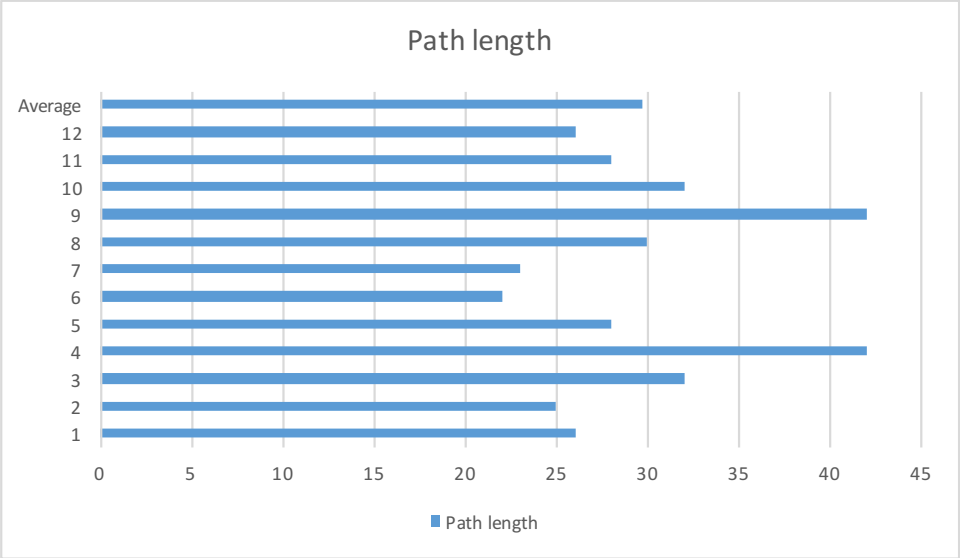


Figure 5.40: Average pathkey length using diagonal-based grouping for a 3×4 network, where $n < m$.

Table 5.40: Number of hops for a 3×4 network using Liu's grouping (**Destination Sensor: 1**).

Node ID	Number of Hops	Path	Number of keys
1	0	0	0
2	1	$2 \rightarrow 1$	1
3	2	$3 \rightarrow 2 \rightarrow 1$	1
4	3	$4 \rightarrow 3 \rightarrow 2 \rightarrow 1$	1
5	1	$5 \rightarrow 1$	1
6	2	$6 \rightarrow 2 \rightarrow 1$	2
7	3	$7 \rightarrow 3 \rightarrow 2 \rightarrow 1$	2
8	4	$8 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1$	2
9	2	$9 \rightarrow 5 \rightarrow 1$	1
10	3	$10 \rightarrow 6 \rightarrow 2 \rightarrow 1$	2
11	4	$11 \rightarrow 7 \rightarrow 3 \rightarrow 2 \rightarrow 1$	2
12	5	$12 \rightarrow 8 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1$	2
Total Hops	30		17

Table 5.41: Number of hops for a 3×4 network using Liu's grouping (**Destination Sensor: 2**).

Node ID	Number of Hops	Path	Number of keys
1	1	$1 \rightarrow 2$	1
2	0	0	0
3	1	$3 \rightarrow 2$	1
4	2	$4 \rightarrow 3 \rightarrow 2$	1
5	2	$5 \rightarrow 1 \rightarrow 2$	2
6	1	$6 \rightarrow 2$	1
7	2	$7 \rightarrow 3 \rightarrow 2$	2
8	3	$8 \rightarrow 4 \rightarrow 3 \rightarrow 2$	2
9	5	$9 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 2$	2
10	4	$10 \rightarrow 6 \rightarrow 2 \rightarrow 3 \rightarrow 2$	1
11	3	$11 \rightarrow 7 \rightarrow 3 \rightarrow 2$	2
12	4	$12 \rightarrow 8 \rightarrow 4 \rightarrow 3 \rightarrow 2$	2
Total Hops	28		17

Table 5.42: Number of hops for a 3×4 network using Liu's grouping (**Destination Sensor: 3**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 2 \rightarrow 3$	1
2	4	$2 \rightarrow 3$	1
3	0	0	0
4	4	$4 \rightarrow 3$	1
5	3	$5 \rightarrow 1 \rightarrow 2 \rightarrow 3$	2
6	2	$6 \rightarrow 2 \rightarrow 3$	2
7	1	$7 \rightarrow 3$	1
8	5	$8 \rightarrow 4 \rightarrow 3$	2
9	4	$9 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3$	2
10	3	$10 \rightarrow 6 \rightarrow 2 \rightarrow 3$	2
11	2	$11 \rightarrow 7 \rightarrow 3$	1
12	3	$12 \rightarrow 8 \rightarrow 4 \rightarrow 3$	2
Total Hops	33		18

Table 5.43: Number of hops for a 3×4 network using Liu's grouping (**Destination Sensor: 4**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	1
2	2	$2 \rightarrow 3 \rightarrow 4$	1
3	1	$3 \rightarrow 4$	1
4	0	0	0
5	4	$5 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	2
6	3	$6 \rightarrow 2 \rightarrow 3 \rightarrow 4$	2
7	2	$7 \rightarrow 3 \rightarrow 4$	2
8	1	$8 \rightarrow 4$	1
9	5	$9 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	2
10	4	$10 \rightarrow 6 \rightarrow 2 \rightarrow 3 \rightarrow 4$	2
11	3	$11 \rightarrow 7 \rightarrow 3 \rightarrow 4$	2
12	2	$12 \rightarrow 8 \rightarrow 4$	1
Total Hops	30		17

Table 5.44: Number of hops for a 3×4 network using Liu's grouping (**Destination Sensor: 5**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 5$	1
2	2	$2 \rightarrow 6 \rightarrow 5$	2
3	3	$3 \rightarrow 7 \rightarrow 6 \rightarrow 5$	2
4	4	$4 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 5$	2
5	0	0	0
6	2	$6 \rightarrow 5$	1
7	2	$7 \rightarrow 6 \rightarrow 5$	1
8	3	$8 \rightarrow 7 \rightarrow 6 \rightarrow 5$	1
9	1	$9 \rightarrow 5$	1
10	2	$10 \rightarrow 6 \rightarrow 5$	2
11	3	$11 \rightarrow 7 \rightarrow 6 \rightarrow 5$	2
12	4	$12 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 5$	2
Total Hops	29		17

Table 5.45: Number of hops for a 3×4 network using Liu's grouping (**Destination Sensor: 6**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 5 \rightarrow 6$	2
2	1	$2 \rightarrow 6$	1
3	2	$3 \rightarrow 7 \rightarrow 6$	2
4	3	$4 \rightarrow 8 \rightarrow 7 \rightarrow 6$	2
5	1	$5 \rightarrow 6$	1
6	0	0	0
7	1	$7 \rightarrow 6$	1
8	2	$8 \rightarrow 7 \rightarrow 6$	1
9	2	$9 \rightarrow 5 \rightarrow 6$	2
10	1	$10 \rightarrow 6$	1
11	2	$11 \rightarrow 7 \rightarrow 6$	2
12	3	$12 \rightarrow 8 \rightarrow 7 \rightarrow 6$	2
Total Hops	20		17

Table 5.46: Number of hops for a 3×4 network using Liu's grouping (**Destination Sensor: 7**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 5 \rightarrow 6 \rightarrow 7$	2
2	2	$2 \rightarrow 6 \rightarrow 7$	2
3	1	$3 \rightarrow 7$	1
4	2	$4 \rightarrow 8 \rightarrow 7$	2
5	2	$5 \rightarrow 6 \rightarrow 7$	1
6	1	$6 \rightarrow 7$	1
7	0	0	0
8	1	$8 \rightarrow 7$	1
9	3	$9 \rightarrow 5 \rightarrow 6 \rightarrow 7$	2
10	2	$10 \rightarrow 6 \rightarrow 7$	2
11	1	$11 \rightarrow 7$	1
12	2	$12 \rightarrow 8 \rightarrow 7$	2
Total Hops	20		17

Table 5.47: Number of hops for a 3×4 network using Liu's grouping (**Destination Sensor: 8**).

Node ID	Number of Hops	Path	Number of keys
1	4	$1 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$	2
2	3	$2 \rightarrow 6 \rightarrow 7 \rightarrow 8$	2
3	2	$3 \rightarrow 7 \rightarrow 8$	2
4	1	$4 \rightarrow 8$	1
5	3	$5 \rightarrow 6 \rightarrow 7 \rightarrow 8$	1
6	2	$6 \rightarrow 7 \rightarrow 8$	1
7	1	$7 \rightarrow 8$	1
8	0	0	0
9	4	$9 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$	2
10	3	$10 \rightarrow 6 \rightarrow 7 \rightarrow 8$	2
11	2	$11 \rightarrow 7 \rightarrow 8$	2
12	1	$12 \rightarrow 8$	1
Total Hops	26		17

Table 5.48: Number of hops for a 3×4 network using Liu's grouping (**Destination Sensor: 9**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 5 \rightarrow 9$	1
2	3	$2 \rightarrow 6 \rightarrow 10 \rightarrow 9$	2
3	4	$3 \rightarrow 7 \rightarrow 11 \rightarrow 10 \rightarrow 9$	2
4	5	$4 \rightarrow 8 \rightarrow 12 \rightarrow 11 \rightarrow 10 \rightarrow 9$	2
5	1	$5 \rightarrow 9$	1
6	5	$6 \rightarrow 10 \rightarrow 9$	2
7	3	$7 \rightarrow 11 \rightarrow 10 \rightarrow 9$	2
8	4	$8 \rightarrow 12 \rightarrow 11 \rightarrow 10 \rightarrow 9$	2
9	0	0	0
10	1	$10 \rightarrow 9$	1
11	2	$11 \rightarrow 10 \rightarrow 9$	1
12	3	$12 \rightarrow 11 \rightarrow 10 \rightarrow 9$	1
Total Hops	33		17

Table 5.49: Number of hops for a 3×4 network using Liu's grouping (**Destination Sensor: 10**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 5 \rightarrow 9 \rightarrow 10$	2
2	2	$2 \rightarrow 6 \rightarrow 10$	1
3	3	$3 \rightarrow 7 \rightarrow 11 \rightarrow 10$	2
4	4	$4 \rightarrow 8 \rightarrow 12 \rightarrow 11 \rightarrow 10$	2
5	2	$5 \rightarrow 9 \rightarrow 10$	2
6	1	$6 \rightarrow 10$	1
7	2	$7 \rightarrow 11 \rightarrow 10$	2
8	3	$8 \rightarrow 12 \rightarrow 11 \rightarrow 10$	2
9	1	$9 \rightarrow 10$	1
10	0	0	0
11	1	$11 \rightarrow 10$	1
12	2	$12 \rightarrow 11 \rightarrow 10$	1
Total Hops	22		17

Table 5.50: Number of hops for a 3×4 network using Liu's grouping (**Destination Sensor: 11**).

Node ID	Number of Hops	Path	Number of keys
1	4	$1 \rightarrow 5 \rightarrow 9 \rightarrow 10 \rightarrow 11$	2
2	3	$2 \rightarrow 6 \rightarrow 10 \rightarrow 11$	2
3	2	$3 \rightarrow 7 \rightarrow 11$	1
4	3	$4 \rightarrow 8 \rightarrow 12 \rightarrow 11$	2
5	3	$5 \rightarrow 9 \rightarrow 10 \rightarrow 11$	2
6	2	$6 \rightarrow 10 \rightarrow 11$	2
7	1	$7 \rightarrow 11$	1
8	2	$8 \rightarrow 12 \rightarrow 11$	2
9	2	$9 \rightarrow 10 \rightarrow 11$	1
10	1	$10 \rightarrow 11$	1
11	0	0	0
12	1	$12 \rightarrow 11$	1
Total Hops	20		17

Table 5.51: Number of hops for a 3×4 network using Liu's grouping (**Destination Sensor: 12**).

Node ID	Number of Hops	Path	Number of keys
1	5	$1 \rightarrow 5 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12$	2
2	4	$2 \rightarrow 6 \rightarrow 10 \rightarrow 11 \rightarrow 12$	2
3	3	$3 \rightarrow 7 \rightarrow 11 \rightarrow 12$	2
4	2	$4 \rightarrow 8 \rightarrow 12$	1
5	4	$5 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12$	2
6	2	$6 \rightarrow 10 \rightarrow 11 \rightarrow 12$	2
7	2	$7 \rightarrow 11 \rightarrow 12$	2
8	1	$8 \rightarrow 12$	1
9	3	$9 \rightarrow 10 \rightarrow 11 \rightarrow 12$	1
10	2	$10 \rightarrow 11 \rightarrow 12$	1
11	1	$11 \rightarrow 12$	1
12	0	0	0
Total Hops	29		17

We can calculate the average pathkey length from the Tables 5.40 to 5.51 for a 3×4 network using Liu's Grouping. This value turns out to be 26.7, as shown in Figure 5.41.

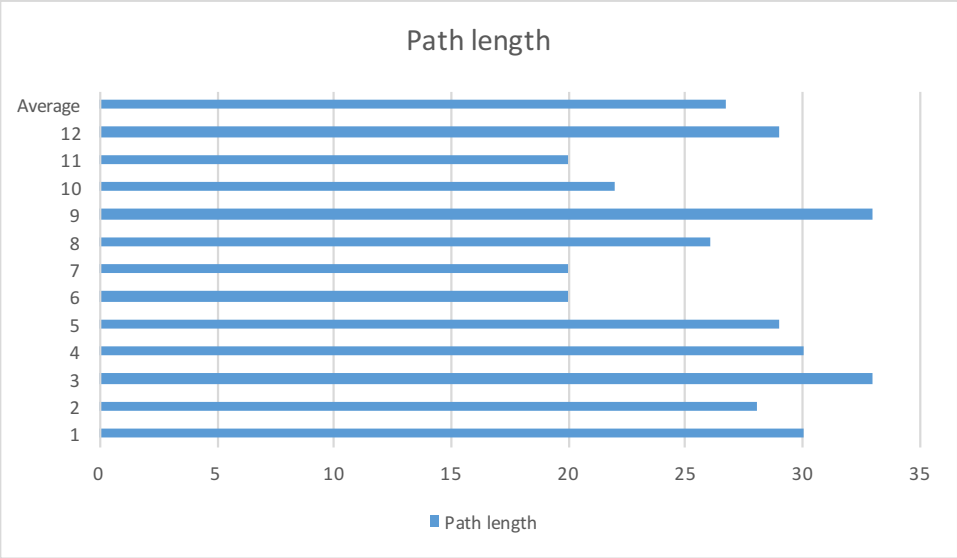


Figure 5.41: Average pathkey length using Liu's Grouping for 3×4 network, where $n < m$.

Table 5.52: Number of hops for a 3×4 network using diagonal$_{\text{min}}$ grouping (**Destination Sensor: 1**).

Node ID	Number of Hops	Path	Number of keys
1	0	0	0
2	1	$2 \rightarrow 1$	1
3	2	$3 \rightarrow 2 \rightarrow 1$	1
4	3	$4 \rightarrow 3 \rightarrow 2 \rightarrow 1$	1
5	4	$5 \rightarrow 10 \rightarrow 11 \rightarrow 6 \rightarrow 1$	3
6	1	$6 \rightarrow 1$	1
7	2	$7 \rightarrow 2 \rightarrow 1$	2
8	3	$8 \rightarrow 3 \rightarrow 2 \rightarrow 1$	2
9	4	$9 \rightarrow 10 \rightarrow 11 \rightarrow 6 \rightarrow 1$	2
10	3	$10 \rightarrow 11 \rightarrow 6 \rightarrow 1$	2
11	2	$11 \rightarrow 6 \rightarrow 1$	1
12	3	$12 \rightarrow 11 \rightarrow 6 \rightarrow 1$	2
Total Hops	28		18

Table 5.53: Number of hops for a 3×4 network using $\text{diagonal}_{\text{min}}$ grouping (**Destination Sensor: 2**).

Node ID	Number of Hops	Path	Number of keys
1	1	$1 \rightarrow 2$	1
2	0	0	0
3	1	$3 \rightarrow 2$	1
4	2	$4 \rightarrow 3 \rightarrow 2$	1
5	5	$5 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 7 \rightarrow 2$	3
6	2	$6 \rightarrow 1 \rightarrow 2$	2
7	1	$7 \rightarrow 2$	1
8	2	$8 \rightarrow 3 \rightarrow 2$	2
9	5	$9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 7 \rightarrow 2$	2
10	4	$10 \rightarrow 11 \rightarrow 12 \rightarrow 7 \rightarrow 2$	2
11	3	$11 \rightarrow 12 \rightarrow 7 \rightarrow 2$	2
12	2	$12 \rightarrow 7 \rightarrow 2$	1
Total Hops	28		18

Table 5.54: Number of hops for a 3×4 network using $\text{diagonal}_{\text{min}}$ grouping (**Destination Sensor: 3**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 2 \rightarrow 3$	1
2	1	$2 \rightarrow 3$	1
3	0	0	0
4	1	$4 \rightarrow 3$	1
5	6	$5 \rightarrow 10 \rightarrow 11 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3$	4
6	3	$6 \rightarrow 1 \rightarrow 2 \rightarrow 3$	2
7	2	$7 \rightarrow 2 \rightarrow 3$	2
8	1	$8 \rightarrow 3$	1
9	6	$9 \rightarrow 10 \rightarrow 11 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3$	3
10	5	$10 \rightarrow 11 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3$	3
11	4	$11 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3$	2
12	3	$12 \rightarrow 7 \rightarrow 2 \rightarrow 3$	2
Total Hops	34		22

Table 5.55: Number of hops for a 3×4 network using diagonal$_{\text{min}}$ grouping (**Destination Sensor: 4**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	1
2	2	$2 \rightarrow 3 \rightarrow 4$	1
3	1	$3 \rightarrow 4$	1
4	0	0	0
5	7	$5 \rightarrow 10 \rightarrow 11 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	4
6	4	$6 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	2
7	3	$7 \rightarrow 2 \rightarrow 3 \rightarrow 4$	2
8	2	$8 \rightarrow 3 \rightarrow 4$	2
9	7	$9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 7 \rightarrow 2 \rightarrow 3 \rightarrow 4$	3
10	6	$10 \rightarrow 11 \rightarrow 12 \rightarrow 7 \rightarrow 2 \rightarrow 3 \rightarrow 4$	3
11	5	$11 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	2
12	4	$12 \rightarrow 7 \rightarrow 2 \rightarrow 3 \rightarrow 4$	2
Total Hops	44		24

Table 5.56: Number of hops for a 3×4 network using diagonal$_{\text{min}}$ grouping (**Destination Sensor: 5**).

Node ID	Number of Hops	Path	Number of keys
1	4	$1 \rightarrow 6 \rightarrow 11 \rightarrow 10 \rightarrow 5$	3
2	5	$2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10 \rightarrow 5$	3
3	6	$3 \rightarrow 2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10 \rightarrow 5$	4
4	7	$4 \rightarrow 3 \rightarrow 2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10 \rightarrow 5$	4
5	0	0	0
6	3	$6 \rightarrow 11 \rightarrow 10 \rightarrow 5$	3
7	4	$7 \rightarrow 12 \rightarrow 11 \rightarrow 10 \rightarrow 5$	3
8	7	$8 \rightarrow 3 \rightarrow 2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10 \rightarrow 5$	5
9	2	$9 \rightarrow 10 \rightarrow 5$	2
10	1	$10 \rightarrow 5$	1
11	2	$11 \rightarrow 10 \rightarrow 5$	2
12	3	$12 \rightarrow 11 \rightarrow 10 \rightarrow 5$	2
Total Hops	44		32

Table 5.57: Number of hops for a 3×4 network using diagonal$_{\text{min}}$ grouping (**Destination Sensor: 6**).

Node ID	Number of Hops	Path	Number of keys
1	1	$1 \rightarrow 6$	1
2	2	$2 \rightarrow 1 \rightarrow 6$	2
3	3	$3 \rightarrow 2 \rightarrow 1 \rightarrow 6$	2
4	4	$4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 6$	2
5	3	$5 \rightarrow 10 \rightarrow 11 \rightarrow 6$	3
6	0	0	0
7	3	$7 \rightarrow 2 \rightarrow 1 \rightarrow 6$	3
8	4	$8 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 6$	3
9	3	$9 \rightarrow 10 \rightarrow 11 \rightarrow 6$	2
10	2	$10 \rightarrow 11 \rightarrow 6$	2
11	1	$11 \rightarrow 6$	1
12	2	$12 \rightarrow 11 \rightarrow 6$	2
Total Hops	28		23

Table 5.58: Number of hops for a 3×4 network using diagonal$_{\text{min}}$ grouping (**Destination Sensor: 7**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 2 \rightarrow 7$	2
2	1	$2 \rightarrow 7$	1
3	2	$3 \rightarrow 2 \rightarrow 7$	2
4	3	$4 \rightarrow 3 \rightarrow 2 \rightarrow 7$	2
5	4	$5 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 7$	3
6	3	$6 \rightarrow 1 \rightarrow 2 \rightarrow 7$	3
7	0	0	0
8	3	$8 \rightarrow 3 \rightarrow 2 \rightarrow 7$	3
9	4	$9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 7$	2
10	3	$10 \rightarrow 11 \rightarrow 12 \rightarrow 7$	2
11	2	$11 \rightarrow 12 \rightarrow 7$	2
12	1	$12 \rightarrow 7$	1
Total Hops	28		23

Table 5.59: Number of hops for a 3×4 network using diagonal$_{\text{min}}$ grouping (**Destination Sensor: 8**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 6 \rightarrow 11 \rightarrow 8$	2
2	2	$2 \rightarrow 3 \rightarrow 8$	2
3	1	$3 \rightarrow 8$	1
4	2	$4 \rightarrow 3 \rightarrow 8$	2
5	7	$5 \rightarrow 10 \rightarrow 11 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 8$	5
6	4	$6 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 8$	3
7	3	$7 \rightarrow 2 \rightarrow 3 \rightarrow 8$	3
8	0	0	0
9	7	$9 \rightarrow 10 \rightarrow 11 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 8$	4
10	6	$10 \rightarrow 5 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 8$	4
11	5	$11 \rightarrow 6 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 8$	3
12	4	$12 \rightarrow 7 \rightarrow 2 \rightarrow 3 \rightarrow 8$	3
Total Hops	40		30

Table 5.60: Number of hops for a 3×4 network using $\text{diagonal}_{\text{min}}$ grouping (**Destination Sensor: 9**).

Node ID	Number of Hops	Path	Number of keys
1	4	$1 \rightarrow 6 \rightarrow 11 \rightarrow 10 \rightarrow 9$	2
2	5	$2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10 \rightarrow 9$	2
3	6	$3 \rightarrow 2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10 \rightarrow 9$	3
4	7	$4 \rightarrow 3 \rightarrow 2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10 \rightarrow 9$	3
5	2	$5 \rightarrow 10 \rightarrow 9$	2
6	3	$6 \rightarrow 11 \rightarrow 10 \rightarrow 9$	2
7	4	$7 \rightarrow 12 \rightarrow 11 \rightarrow 10 \rightarrow 9$	2
8	7	$8 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 6 \rightarrow 11 \rightarrow 10 \rightarrow 9$	4
9	0	0	0
10	1	$10 \rightarrow 9$	1
11	2	$11 \rightarrow 10 \rightarrow 9$	1
12	3	$12 \rightarrow 11 \rightarrow 10 \rightarrow 9$	1
Total Hops	44		23

Table 5.61: Number of hops for a 3×4 network using diagonal$_{\text{min}}$ grouping (**Destination Sensor: 10**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 6 \rightarrow 11 \rightarrow 10$	2
2	4	$2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10$	2
3	5	$3 \rightarrow 2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10$	3
4	6	$4 \rightarrow 3 \rightarrow 2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10$	3
5	1	$5 \rightarrow 10$	1
6	2	$6 \rightarrow 11 \rightarrow 10$	2
7	3	$7 \rightarrow 12 \rightarrow 11 \rightarrow 10$	2
8	6	$8 \rightarrow 3 \rightarrow 2 \rightarrow 7 \rightarrow 12 \rightarrow 11 \rightarrow 10$	3
9	1	$9 \rightarrow 10$	1
10	0	0	0
11	1	$11 \rightarrow 10$	1
12	2	$12 \rightarrow 11 \rightarrow 10$	1
Total Hops	34		21

Table 5.62: Number of hops for a 3×4 network using diagonal$_{\text{min}}$ grouping (**Destination Sensor: 11**).

Node ID	Number of Hops	Path	Number of keys
1	2	$1 \rightarrow 6 \rightarrow 11$	1
2	3	$2 \rightarrow 7 \rightarrow 12 \rightarrow 11$	2
3	4	$3 \rightarrow 2 \rightarrow 7 \rightarrow 12 \rightarrow 11$	3
4	5	$4 \rightarrow 3 \rightarrow 2 \rightarrow 7 \rightarrow 12 \rightarrow 11$	3
5	4	$5 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 11$	2
6	1	$6 \rightarrow 11$	1
7	2	$7 \rightarrow 12 \rightarrow 11$	2
8	5	$8 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 6 \rightarrow 11$	3
9	2	$9 \rightarrow 10 \rightarrow 11$	1
10	1	$10 \rightarrow 11$	1
11	0	0	0
12	1	$12 \rightarrow 11$	1
Total Hops	30		20

Table 5.63: Number of hops for a 3×4 network using diagonal$_{\text{min}}$ grouping (**Destination Sensor: 12**).

Node ID	Number of Hops	Path	Number of keys
1	3	$1 \rightarrow 6 \rightarrow 11 \rightarrow 12$	2
2	2	$2 \rightarrow 7 \rightarrow 12$	1
3	3	$3 \rightarrow 2 \rightarrow 7 \rightarrow 12$	2
4	4	$4 \rightarrow 3 \rightarrow 2 \rightarrow 7 \rightarrow 12$	2
5	3	$5 \rightarrow 10 \rightarrow 11 \rightarrow 12$	2
6	2	$6 \rightarrow 11 \rightarrow 12$	2
7	1	$7 \rightarrow 12$	1
8	4	$8 \rightarrow 3 \rightarrow 2 \rightarrow 7 \rightarrow 12$	3
9	3	$9 \rightarrow 10 \rightarrow 11 \rightarrow 12$	1
10	2	$10 \rightarrow 11 \rightarrow 12$	1
11	1	$11 \rightarrow 12$	1
12	0	0	0
Total Hops	29		18

We can calculate the average pathkey length the Tables 5.52 to 5.63 for a 3×4 network using diagonal$_{\text{min}}$ grouping. This value turns out to be 35, as shown in Figure 5.42.

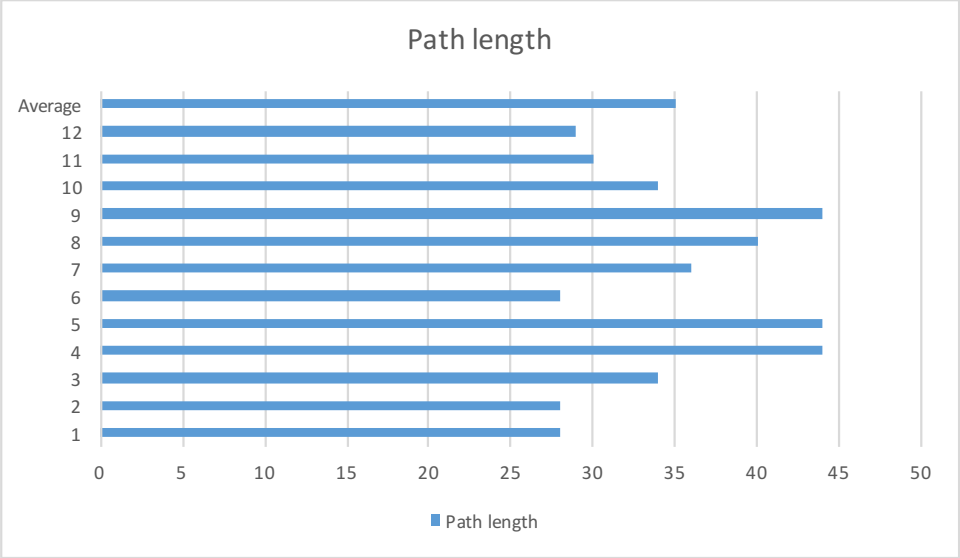


Figure 5.42: Average pathkey length using diagonal$_{\text{min}}$ grouping for 3×4 network, where $n < m$.

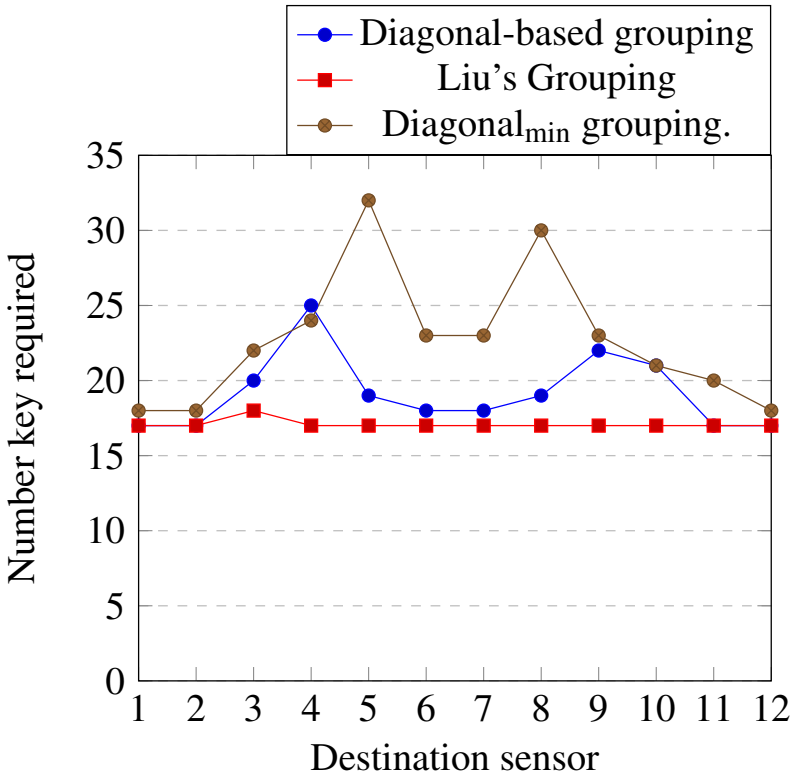


Figure 5.43: Comparison of number of keys utilized.

As shown in Figure 5.43 the proposed grouping performs close to Liu's grouping when there are lots of transmissions along the diagonal direction. Sensor 4 and 9 do not belong to any diagonal. That is why their required keys are high. For diagonal$_{min}$, sensor 5 and 8 do not belong to any rows resulting in high key requirement.

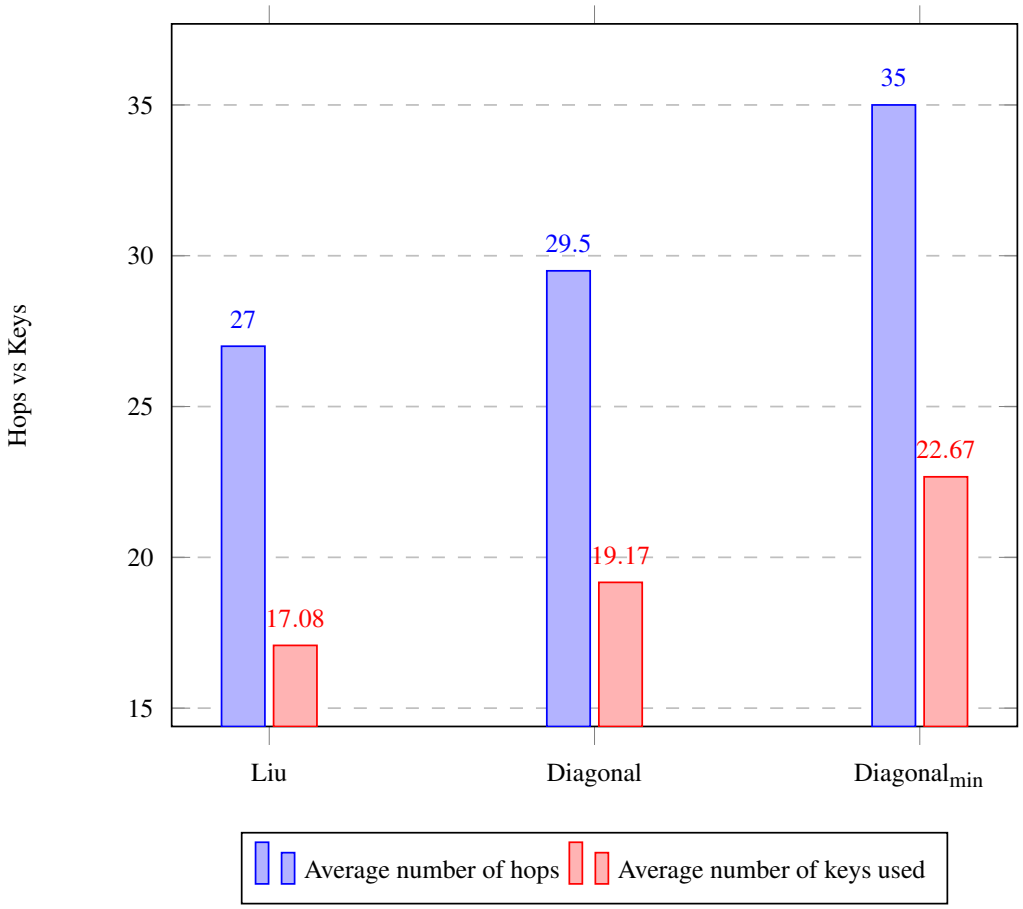


Figure 5.44: Comparison between average number of hops and average number of keys used 3×4 network, where $n < m$.

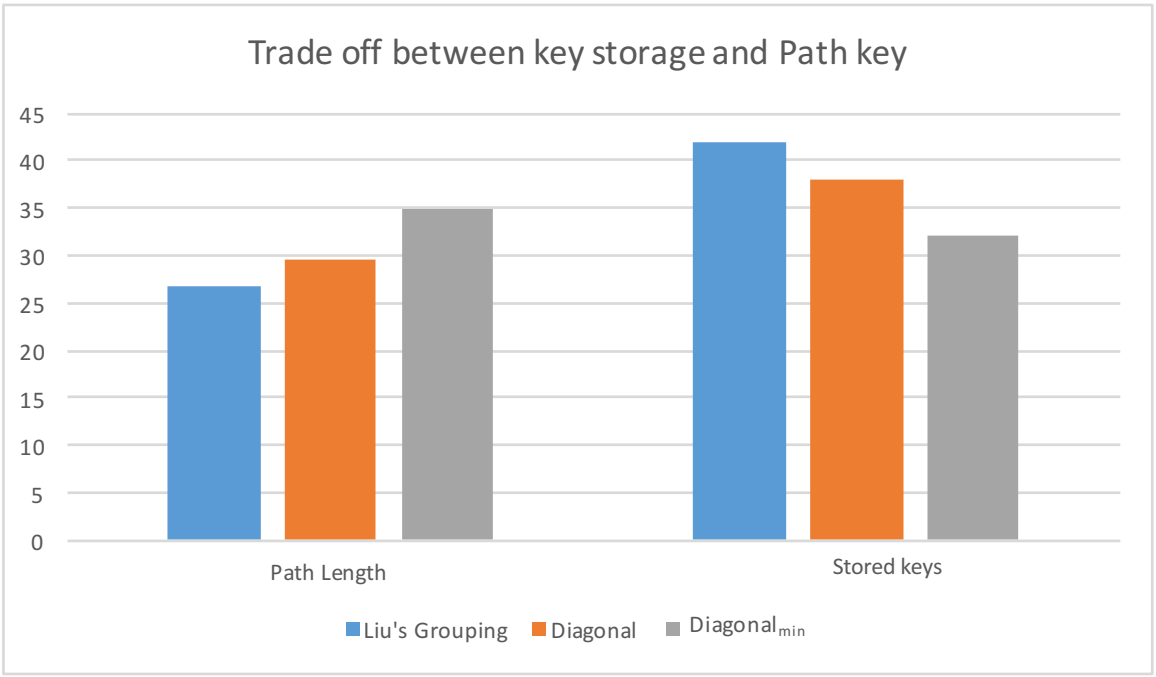


Figure 5.45: Trade-off between different groupings or 3×4 network, where $n < m$.

For a 3×4 network, where $n < m$, the proposed diagonal-based grouping performs close to Liu's grouping as shown in Figure 5.44 and the key utilization of the proposed grouping is also close to Liu's grouping as shown in Figure 5.44 .

We next calculate the number of hops for a 6×5 network, where $n > m$ using Liu's grouping, diagonal-based grouping, and diagonal$_{\min}$ grouping respectively. Figure 5.46 illustrates an example of the diagonal-based grouping implementation.

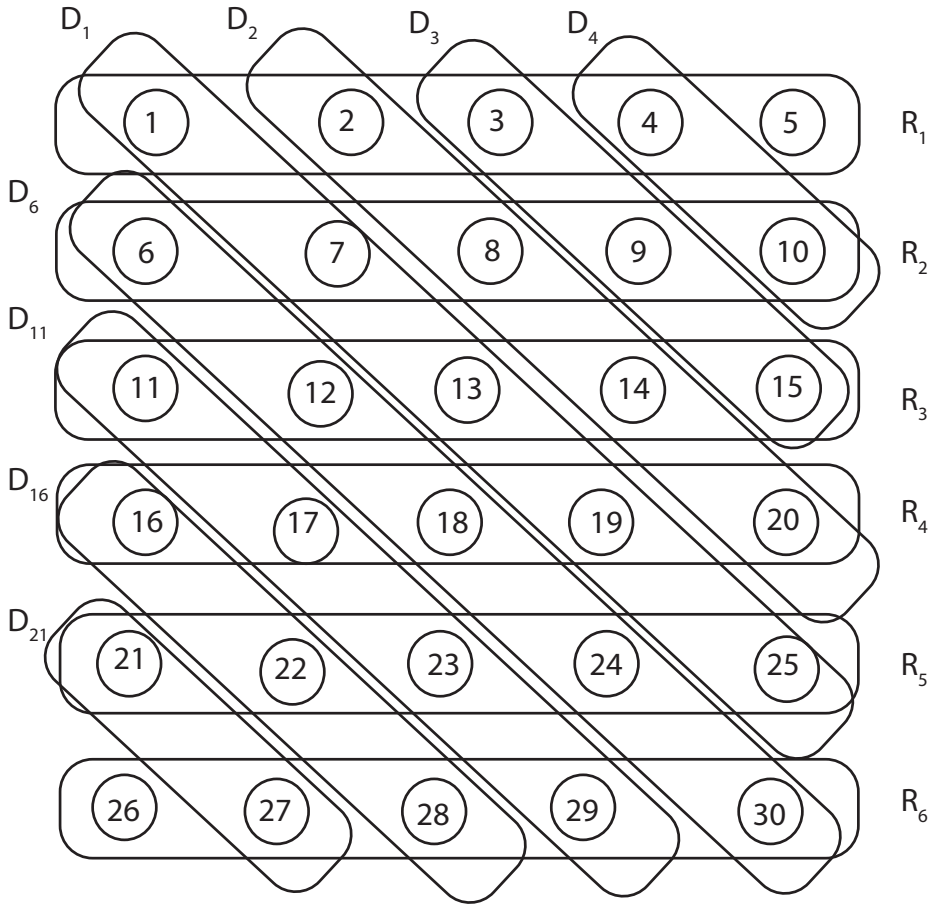


Figure 5.46: A 6×5 network, where $n > m$ using diagonal-based grouping.

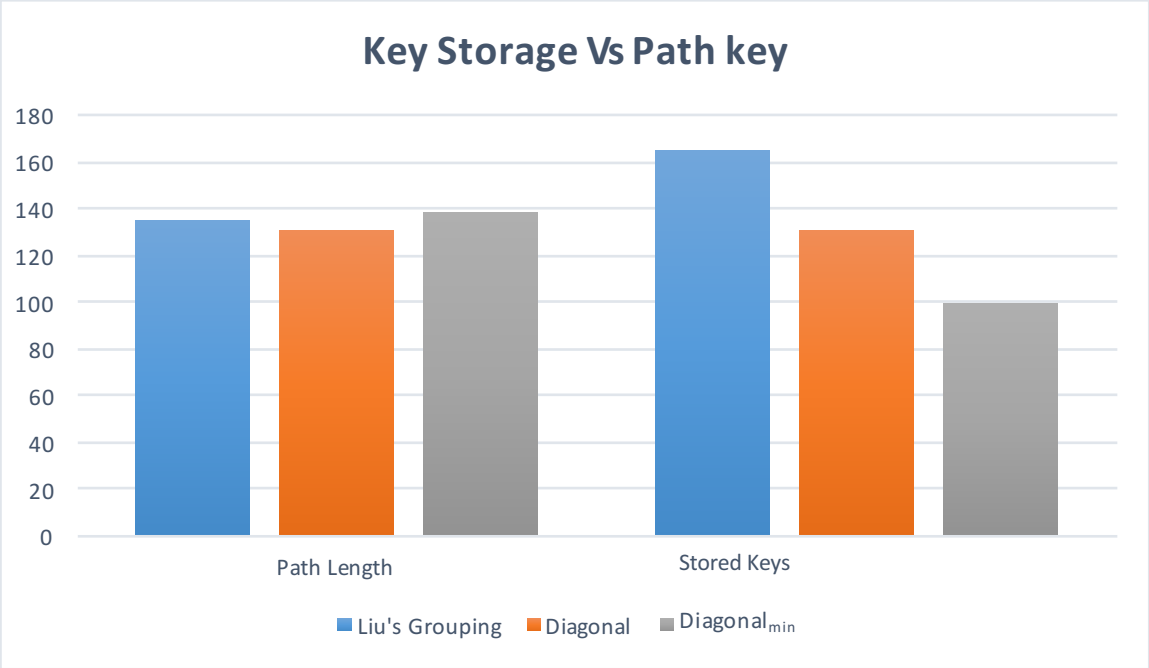


Figure 5.47: Trade-off between different grouping for 6×5 network, where $n > m$.

Fora 6×5 network, where $n > m$, the proposed diagonal-based grouping performs significantly better than Liu's grouping as shown in Figure 5.47.

Thus we can conclude that the performance of these groups changes based on the network orientations. However, Liu's grouping also requires a high amount of key to store. The proposed diagonal-based grouping and $\text{diagonal}_{\text{min}}$ grouping requires 15% and 26% fewer key respectively. The key storage is permanent where the path is only used when sensors are trying to communicate. This amount varies depending on the nature of the task given to the sensor. Beside this, there are multiple paths to reach any sensors in the network, meaning certain path may not be utilized at all. But key storage will be the same no matter what the situation is.

Based on the observations from the pathkey length a general mathematical formula to calculate diagonal groups (D_a) is described as follows:

$$\sum_{a=min}^{a=max} \underbrace{p}_{\text{total number of } D_a} \cdot \underbrace{(n-1)}_{\text{length of Da}}$$

Table 5.64: Number of hops for a 6×5 network using diagonal-based grouping (**Destination Sensor: 30**).

Node ID	Number of Hops	Path	Number of keys
1	6	$1 \rightarrow 7 \rightarrow 13 \rightarrow 19 \rightarrow 25 \rightarrow 24 \rightarrow 30$	3
2	7	$2 \rightarrow 8 \rightarrow 14 \rightarrow 20 \rightarrow 19 \rightarrow 25 \rightarrow 24 \rightarrow 30$	3
3	8	$3 \rightarrow 9 \rightarrow 15 \rightarrow 14 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	3
4	9	$4 \rightarrow 10 \rightarrow 9 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	3
5	10	$5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 7 \rightarrow 13 \rightarrow 19 \rightarrow 25 \rightarrow 24 \rightarrow 30$	4
6	4	$6 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	1
7	5	$7 \rightarrow 13 \rightarrow 19 \rightarrow 25 \rightarrow 24 \rightarrow 30$	2
8	6	$8 \rightarrow 7 \rightarrow 6 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	2
9	7	$9 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	2
10	8	$10 \rightarrow 9 \rightarrow 8 \rightarrow 7 \rightarrow 6 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	2
11	4	$11 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	2
12	3	$12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	1
13	4	$13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	2
14	5	$14 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	2
15	6	$15 \rightarrow 14 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	2
16	4	$16 \rightarrow 17 \rightarrow 18 \rightarrow 24 \rightarrow 30$	2
17	3	$17 \rightarrow 18 \rightarrow 24 \rightarrow 30$	2
18	2	$18 \rightarrow 24 \rightarrow 30$	1
19	3	$19 \rightarrow 18 \rightarrow 24 \rightarrow 30$	2
20	4	$20 \rightarrow 19 \rightarrow 18 \rightarrow 24 \rightarrow 30$	2
21	4	$21 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
22	3	$22 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
23	2	$23 \rightarrow 29 \rightarrow 30$	2
24	1	$24 \rightarrow 30$	1
25	2	$25 \rightarrow 24 \rightarrow 30$	2
26	4	$26 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	1
27	3	$27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	1
28	2	$28 \rightarrow 29 \rightarrow 30$	1
29	1	$29 \rightarrow 30$	1
30	0	0	0
Total Hops	130		56

Table 5.65: Number of hops for a 6×5 network using $\text{diagonal}_{\text{min}}$ grouping (**Destination Sensor: 30**).

Node ID	Number of Hops	Path	Number of keys
1	6	$1 \rightarrow 7 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	3
2	7	$2 \rightarrow 8 \rightarrow 14 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	3
3	8	$3 \rightarrow 9 \rightarrow 15 \rightarrow 14 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	3
4	9	$4 \rightarrow 3 \rightarrow 9 \rightarrow 15 \rightarrow 14 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	4
5	10	$5 \rightarrow 4 \rightarrow 3 \rightarrow 9 \rightarrow 15 \rightarrow 14 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	4
6	4	$6 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	1
7	5	$7 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	3
8	6	$8 \rightarrow 14 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	3
9	7	$9 \rightarrow 15 \rightarrow 14 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	3
10	8	$10 \rightarrow 4 \rightarrow 3 \rightarrow 9 \rightarrow 15 \rightarrow 14 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	5
11	4	$11 \rightarrow 17 \rightarrow 23 \rightarrow 29 \rightarrow 30$	2
12	3	$12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	1
13	4	$13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	2
14	5	$14 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	2
15	6	$15 \rightarrow 14 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	2
16	4	$16 \rightarrow 22 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
17	3	$17 \rightarrow 23 \rightarrow 29 \rightarrow 30$	2
18	2	$18 \rightarrow 24 \rightarrow 30$	1
19	5	$19 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	3
20	6	$20 \rightarrow 14 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	3
21	4	$21 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
22	3	$22 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
23	2	$23 \rightarrow 29 \rightarrow 30$	2
24	1	$24 \rightarrow 30$	1
25	6	$25 \rightarrow 19 \rightarrow 13 \rightarrow 12 \rightarrow 18 \rightarrow 24 \rightarrow 30$	3
26	4	$26 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	1
27	3	$27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	1
28	2	$28 \rightarrow 29 \rightarrow 30$	1
29	1	$29 \rightarrow 30$	1
30	0	0	0
Total Hops	138		66

Table 5.66: Number of hops for a 6×5 network using Liu's grouping (**Destination Sensor: 30**).

Node ID	Number of Hops	Path	Number of keys
1	9	$1 \rightarrow 6 \rightarrow 11 \rightarrow 16 \rightarrow 21 \rightarrow 26 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
2	8	$2 \rightarrow 7 \rightarrow 12 \rightarrow 17 \rightarrow 22 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
3	7	$3 \rightarrow 8 \rightarrow 13 \rightarrow 18 \rightarrow 23 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
4	6	$4 \rightarrow 9 \rightarrow 14 \rightarrow 19 \rightarrow 24 \rightarrow 29 \rightarrow 30$	2
5	5	$5 \rightarrow 10 \rightarrow 15 \rightarrow 20 \rightarrow 25 \rightarrow 30$	1
6	8	$6 \rightarrow 11 \rightarrow 16 \rightarrow 21 \rightarrow 26 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
7	7	$7 \rightarrow 12 \rightarrow 17 \rightarrow 22 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
8	6	$8 \rightarrow 13 \rightarrow 18 \rightarrow 23 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
9	5	$9 \rightarrow 14 \rightarrow 19 \rightarrow 24 \rightarrow 29 \rightarrow 30$	2
10	4	$10 \rightarrow 15 \rightarrow 20 \rightarrow 25 \rightarrow 30$	1
11	7	$11 \rightarrow 16 \rightarrow 21 \rightarrow 26 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
12	6	$12 \rightarrow 17 \rightarrow 22 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
13	5	$13 \rightarrow 18 \rightarrow 23 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
14	4	$14 \rightarrow 19 \rightarrow 24 \rightarrow 29 \rightarrow 30$	2
15	3	$15 \rightarrow 20 \rightarrow 25 \rightarrow 30$	1
16	6	$16 \rightarrow 21 \rightarrow 26 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
17	5	$17 \rightarrow 22 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
18	4	$18 \rightarrow 23 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
19	3	$19 \rightarrow 24 \rightarrow 29 \rightarrow 30$	2
20	2	$20 \rightarrow 25 \rightarrow 30$	1
21	5	$21 \rightarrow 26 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
22	4	$22 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
23	3	$23 \rightarrow 28 \rightarrow 29 \rightarrow 30$	2
24	2	$24 \rightarrow 29 \rightarrow 30$	2
25	1	$25 \rightarrow 30$	1
26	4	$26 \rightarrow 27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	1
27	3	$27 \rightarrow 28 \rightarrow 29 \rightarrow 30$	1
28	2	$28 \rightarrow 29 \rightarrow 30$	1
29	1	$29 \rightarrow 30$	1
30	0	0	0
Total Hops	135		49

The general formula is as follow for different network orientations.

1. For $n = m$ network,

$$(n-m+1)(n-1)+\sum_{a=2}^{a=(m-1)} 2(m-a) \tag{5.1}$$

2. For $n > m$ network,

$$(n-m+1)(m-1)\sum_{a=2}^{a=(m-1)} 2(m-a) \tag{5.2}$$

3. For $n < m$ network,

$$(m-n+1)(n-1)\sum_{a=2}^{a=(n-1)} 2(n-a) \tag{5.3}$$

As an example let us apply these formulas to the networks as shown in Figure 5.48.

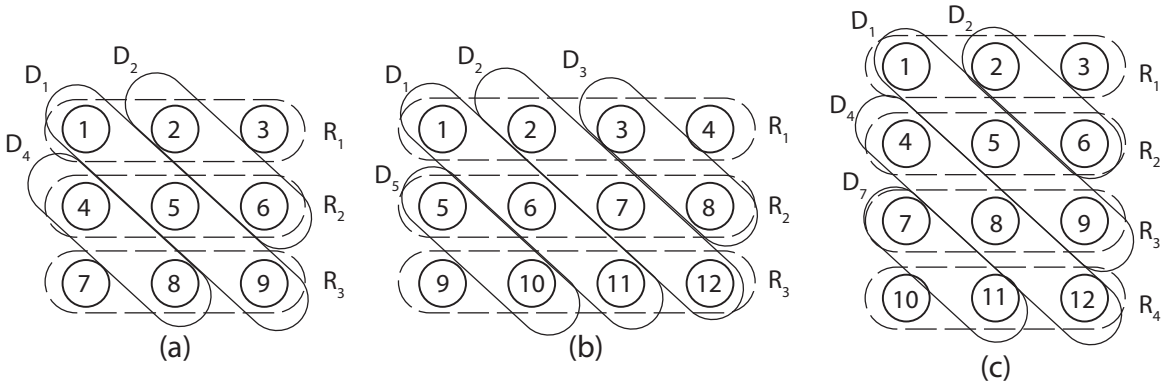


Figure 5.48: Diagonal group instantiation on different types of network orientation.

1. Applying formula 5.1 to the network as shown in Figure 5.48 (a) where $n = m$:

Here, $n = 3$ and $m = 3$

$$= (n-m+1)(n-1) + \sum_{a=2}^{a=(m-1)} 2(m-a)$$
$$= 1.(3-1) + \sum_{a=2}^{a=2} 2(3-a)$$
$$= 1.2 + 2.1$$

This shows us that there is one diagonal group (D_a) with size 2 and two diagonal groups (D_a) with size 1. So, there are total three D_a for this network.

2. Applying formula 5.2 to the network as shown in Figure 5.48 (c) where $n > m$

   Here, $n = 4$ and $m = 3$

$$= (n-m+1)(m-1) \sum_{a=2}^{a=(m-1)} 2(m-a)$$
$$= 2.(3-1) \sum_{a=2}^{a=2} 2(3-a)$$
$$= 2.2 + 2.1$$

   This tells us there are two diagonal groups (D_a) with size 2 and two diagonal groups (D_a) with size 1.

3. Applying formula 5.3 to the network as shown in Figure 5.48 (b) where $n < m$

   Here, $n = 3$ and $m = 4$

$$= (m-n+1)(n-1) \sum_{a=2}^{a=(n-1)} 2(n-a)$$
$$= 2.(3-1) \sum_{a=2}^{a=2} 2(3-a)$$
$$= 2.2 + 2.1$$

   This tells us there are two diagonal groups (D_a) with size 2 and two diagonal groups

(D_a) with size 1.

One of the advantage of diagonal-based grouping or diagonal$_{\text{min}}$ grouping is that across the diagonal any node requires only one key while Liu's grouping always requires 2 keys (one column and one row) to encrypt data as shown in Table 5.67.

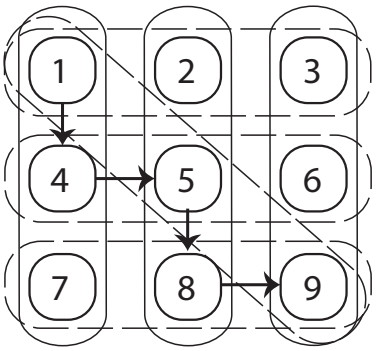


Figure 5.49: Key encryption for Liu's grouping across diagonal.

Using the above formulas and algorithm discussed in Section 4.4 comparisons between diagonal-based grouping and Liu's grouping for diagonals are shown in Figure 5.50 and Table 5.67. The values across diagonal groups are the same for diagonal-based and diagonal$_{\text{min}}$ grouping.

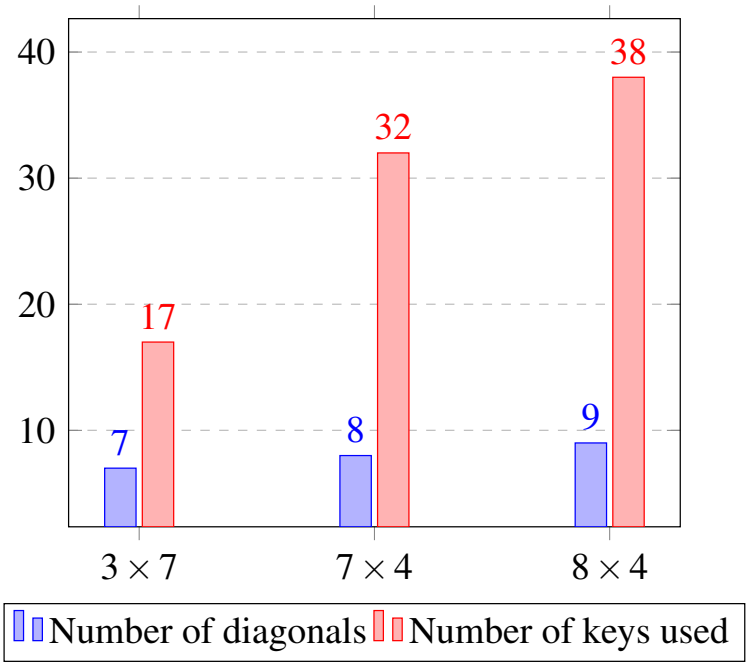


Figure 5.50: Comparison between number of diagonals and number of keys used on diagonal-based grouping for different $n \times m$ network.

Table 5.67: Number of keys required for a 3×3 network using diagonal-based and Liu's grouping.

Path across diagonal	Number of keys for diagonal-based grouping	Number of keys for Liu's grouping
$1 \rightarrow 5$	1	2
$1 \rightarrow 9$	1	2
$5 \rightarrow 9$	1	2
$2 \rightarrow 6$	1	2
$4 \rightarrow 8$	1	2

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The use of WSN has increased significantly over the last decade. Sensors are now being used in large numbers almost everywhere. This has posed a major challenge for efficient grouping schemes. There has to be a balance between storage capacity and routing protocol. This was the main motivation for this thesis. Many notable works related to this issue have been performed but none of them explored the concept of diagonals in WSN. Therefore, we developed a diagonal-based grouping to improve the security and performance of key distribution based on the work done by Liu, Ning, and Du [1]. The reason for choosing their work was because Liu et al. argued that the sensor nodes in the same group generally reside close to each other after deployment, and did not assume prior knowledge of any deployment points as was the case in some previous research [46, 51] in which deployment points were pre-determined. Liu et al. [1] divided the network into rows and columns, whereas the proposed grouping is based on rows and diagonals. Thus, two different types of grouping schemes were developed, diagonal grouping and $\text{diagonal}_{\text{min}}$ grouping.

Although diagonal-based and $\text{diagonal}_{\text{min}}$ grouping have the same underlying structure, the $\text{diagonal}_{\text{min}}$ requires a fewer number of rows. However, $\text{diagonal}_{\text{min}}$ grouping often requires more hops to reach the destination, although the required key storage is significantly less than it is for the other two schemes. If a network is required to communicate less often but stay fully connected, the $\text{diagonal}_{\text{min}}$ grouping is the right choice. Diagonal grouping is a balance between both key storage and pathkey length. It is suitable for any kind of

network orientations. At the end of the thesis, the following goals have been achieved:

1. **New Grouping Scheme:** The most significant contribution of the thesis is the introduction of diagonal-based grouping. While traditional grouping schemes have considered different types of groupings, such as squares, columns, and circles, we are not aware of any previous groupings based on diagonals.

2. **Key Storage:** Researchers are continuously striving to find better solutions for storage-constrained sensors. Thus, researchers had to be consider the limited storage capacity when implementing new grouping schemes. diagonal-based grouping and $\text{diagonal}_{\text{min}}$ grouping has shown a significant reduction in the number of keys required.

3. **Hash key based communication:** The sensors in the diagonal groups can identify the neighbour nodes easily in order to communicate. I have discussed the implementation and distribution of hash keys extensively in this thesis. This technique saves redundant communication while establishing a pairwise connection.

4. **Network Orientation:** Most of the implementation related to WSN did not consider how the grouping works if the network orientation changes. This thesis has also considered this important factor and has demonstrated that the performance of groupings varies according to network orientations. Three different network orientations were considered:

   (a) A network with equal rows and columns where the number of rows (n) and sensors in each row (m); that is, $n = m$. Example: 3×3, 4×4, 5×5, or 6×6 networks.

   (b) A network with fewer rows than columns; that is, where $n < m$. Example: 3×4, 3×7, 4×5, 4×11, 5×6, or 6×7 networks. and

(c) A network with more rows than columns; that is, $n > m$. Example: 4×3, 7×4, 5×4, 6×5, or 12×6 networks.

I believe this thesis has created an opportunity to explore an alternative way of looking at how grouping is done in WSN. This grouping scheme might inspire future researchers to develop a complete framework by exploiting the concept of the diagonal.

6.2 Limitations

Although the proposed grouping requires less key storage than Liu's grouping [1], it requires more paths. However, if there is a lot of data flow across diagonal, the proposed grouping demonstrates lower pathkey length with efficient key utilization.

6.3 Future Work

The implementation of diagonal-based grouping has lot of scope. Diagonal-based grouping could be introduced into various types of network. One of the challenging application could be the development of diagonal-based routing protocol.

A mathematical model of diagonal-based grouping has been developed in this thesis. This grouping could also be implemented in combination with Liu's grouping [1] in different network orientations.

A hash-key-based pairwise scheme is used in the proposed grouping. Researchers could explore other available schemes for key distribution, such as probabilistic method or elliptic curve cryptography using the proposed grouping scheme.

The proposed diagonal-based grouping has been implemented for a maximum network size of four hundred. In the future, the scalability of this scheme could be tested by adding more sensor nodes to the network.

Bibliography

[1] Donggang Liu, Peng Ning, and Wenliang Du. Group-based key predistribution for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 4(2):11, 2008.

[2] William Stallings. *Cryptography and Network Security, 4/E*. Pearson Education India, 2006.

[3] Khan Md Asif, Tamim Israfil, Ahmed Emdad, and Awal M Abdul. Multiple parameter based clustering (mpc): Prospective analysis for effective clustering in wireless sensor network (wsn) using k-means algorithm. *Wireless Sensor Network*, 2012, 2011.

[4] Manijeh Keshtgari and Amene Deljoo. A wireless sensor network solution for precision agriculture based on zigbee technology. *Wireless Sensor Network*, 4(1):25, 2012.

[5] Michael Winkler, Michael Street, Klaus-Dieter Tuchs, and Konrad Wrona. Wireless sensor networks for military purposes. In *Autonomous Sensor Networks*, pages 365–394. Springer, 2012.

[6] Amir Hoshang Kioumars and Liqiong Tang. Wireless network for health monitoring: Heart rate and temperature sensor. In *Sensing Technology (ICST), 2011 Fifth International Conference on*, pages 362–369. IEEE, 2011.

[7] Gianluigi Ferrari, Paolo Medagliani, S Di Piazza, and Marco Martalò. Wireless sensor networks: performance analysis in indoor scenarios. *EURASIP Journal on Wireless Communications and Networking*, 2007(1):41–41, 2007.

[8] Fei Ding, Guangming Song, Kaijian Yin, Jianqing Li, and Aiguo Song. A gps-enabled wireless sensor network for monitoring radioactive materials. *Sensors and Actuators A: Physical*, 155(1):210–215, 2009.

[9] Laurent Eschenauer and Virgil D Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47. ACM, 2002.

[10] Haowen Chan, Adrian Perrig, and Dawn Song. Random key predistribution schemes for sensor networks. In *Security and Privacy, 2003. Proceedings. 2003 Symposium on*, pages 197–213. IEEE, 2003.

[11] Haowen Chan and Adrian Perrig. Pike: Peer intermediaries for key establishment in sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 1, pages 524–535. IEEE, 2005.

[12] Donggang Liu and Peng Ning. Location-based pairwise key establishments for static sensor networks. In *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 72–82. ACM, 2003.

[13] Fan Ye, Haiyun Luo, Jerry Cheng, Songwu Lu, and Lixia Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 148–159. ACM, 2002.

[14] Fan Ye, Alvin Chen, Songwu Lu, and Lixia Zhang. A scalable solution to minimum cost forwarding in large sensor networks. In *Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on*, pages 304–309. IEEE, 2001.

[15] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185. ACM, 1999.

[16] Joanna Kulik, Wendi Heinzelman, and Hari Balakrishnan. Negotiation-based protocols for disseminating information in wireless sensor networks. *Wireless networks*, 8(2/3):169–185, 2002.

[17] Jianmin Zhang, Jianwei Tan, and Jian Li. Key distribution using double keyed-hash chains for wireless sensor networks. *International Journal of Security & Its Applications*, 7(5), 2013.

[18] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):11–25, 2001.

[19] Minghui Shi, Xuemin Shen, Yixin Jiang, and Chuang Lin. Self-healing group-wise key distribution schemes with time-limited node revocation for wireless sensor networks. *Wireless Communications, IEEE*, 14(5):38–46, 2007.

[20] Whitfield Diffie and Martin E Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976.

[21] Ronald L Rivest, Adi Shamir, and Len Adleman. A method for obtaining digital

signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[22] Michael Brown, Donny Cheung, Darrel Hankerson, Julio Lopez Hernandez, Michael Kirkup, and Alfred Menezes. Pgp in constrained wireless devices. In *USENIX Security Symposium*, 2000.

[23] David W Carman, Peter S Kruus, and Brian J Matt. Constraints and approaches for distributed sensor network security (final). *DARPA Project report,(Cryptographic Technologies Group, Trusted Information System, NAI Labs)*, 1(1), 2000.

[24] Anthony D Wood and John A Stankovic. Denial of service in sensor networks. *Computer*, 35(10):54–62, 2002.

[25] Jason L Hill and David E Culler. Mica: A wireless platform for deeply embedded networks. *Micro, IEEE*, 22(6):12–24, 2002.

[26] Lidong Zhou and Zygmunt J Haas. Securing ad hoc networks. *Network, IEEE*, 13(6):24–30, 1999.

[27] Frank Stajano and Ross Anderson. The cocaine auction protocol: On the power of anonymous broadcast. In *Information Hiding*, pages 434–447. Springer, 1999.

[28] Seyit A Camtepe and Bülent Yener. Key distribution mechanisms for wireless sensor networks: a survey. *Rensselaer Polytechnic Institute, Troy, New York, Technical Report*, pages 05–07, 2005.

[29] Sushmita Ruj, Amiya Nayak, and Ivan Stojmenovic. Pairwise and triple key distribution in wireless sensor networks with applications. *Computers, IEEE Transactions on*, 62(11):2224–2237, 2013.

[30] Sushmita Ruj, Amiya Nayak, and Ivan Stojmenovic. Fully secure pairwise and triple key distribution in wireless sensor networks using combinatorial designs. In *INFOCOM, 2011 Proceedings IEEE*, pages 326–330. IEEE, 2011.

[31] B Clifford Neuman and Theodore Ts' O. Kerberos: An authentication service for computer networks. *Communications Magazine, IEEE*, 32(9):33–38, 1994.

[32] Adrian Perrig, Robert Szewczyk, Justin Douglas Tygar, Victor Wen, and David E Culler. Spins: Security protocols for sensor networks. *Wireless networks*, 8(5):521–534, 2002.

[33] C-S Laih, M-K Sun, C-C Chang, and Y-S Han. Adaptive key pre-distribution model for distributed sensor networks. *Communications, IET*, 3(5):723–732, 2009.

[34] PUB FIPS. 46-3: Data encryption standard (des). *National Institute of Standards and Technology*, 25(10):1–22, 1999.

[35] Jiang Jian-wei et al. Research on key management scheme for wsn based on elliptic curve cryptosystem. In *Networked Digital Technologies, 2009. NDT'09. First International Conference on*, pages 536–540. IEEE, 2009.

[36] Chien-Wen Chiang, Chih-Chung Lin, and Ray-I Chang. A new scheme of key distribution using implicit security in wireless sensor networks. In *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, volume 1, pages 151–155. IEEE, 2010.

[37] Li Zhou, Jinfeng Ni, and Chinya V Ravishankar. Efficient key establishment for group-based wireless sensor deployments. In *Proceedings of the 4th ACM workshop on Wireless security*, pages 1–10. ACM, 2005.

[38] Lei Yu and Jianzhong Li. Grouping-based resilient statistical en-route filtering for sensor networks. In *INFOCOM 2009, IEEE*, pages 1782–1790. IEEE, 2009.

[39] Dnyaneshwar Mantri, Neeli Rashmi Prasad, and Ranga Prasad. Grouping of clusters for efficient data aggregation (gceda) in wireless sensor network. In *Advance Computing Conference (IACC), 2013 IEEE 3rd International*, pages 132–137. IEEE, 2013.

[40] Lein Harn and Ching-Fang Hsu. Predistribution scheme for establishing group keys in wireless sensor networks. *Sensors Journal, IEEE*, 15(9):5103–5108, 2015.

[41] Eleni Klaoudatou, Elisavet Konstantinou, Georgios Kambourakis, and Stefanos Gritzalis. A survey on cluster-based group key agreement protocols for wsns. *Communications Surveys & Tutorials, IEEE*, 13(3):429–442, 2011.

[42] Katerina Simonova, Alan CH Ling, and X Sean Wang. Location-aware key predistribution scheme for wide area wireless sensor networks. In *Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*, pages 157–168. ACM, 2006.

[43] Cungang Yang and Jie Xiao. Location-based pairwise key establishment and data authentication for wireless sensor networks. In *Information Assurance Workshop, 2006 IEEE*, pages 247–252. IEEE, 2006.

[44] Wenliang Du, Jing Deng, Yunghsiang S Han, Shigang Chen, and Prainod K Varshney. A key management scheme for wireless sensor networks using deployment knowledge. In *INFOCOM 2004. Twenty-third AnnualJoint conference of the IEEE computer and communications societies*, volume 1. IEEE, 2004.

[45] Donggang Liu, Peng Ning, and Rongfang Li. Establishing pairwise keys in distributed sensor networks. *ACM Transactions on Information and System Security (TISSEC)*, 8(1):41–77, 2005.

[46] Dijiang Huang, Manish Mehta, Deep Medhi, and Lein Harn. Location-aware key management scheme for wireless sensor networks. In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 29–42. ACM, 2004.

[47] Dijiang Huang and Deep Medhi. Secure pairwise key establishment in large-scale sensor networks: An area partitioning and multigroup key predistribution approach. *ACM Transactions on Sensor Networks (TOSN)*, 3(3):16, 2007.

[48] Zhen Yu and Yong Guan. A key management scheme using deployment knowledge for wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 19(10):1411–1425, 2008.

[49] Mohamed F Younis, Kajaldeep Ghumman, and Mohamed Eltoweissy. Location-aware combinatorial key management scheme for clustered sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 17(8):865–882, 2006.

[50] Dahai Xu, Jianwei Huang, Jeffrey Dwoskin, Mung Chiang, and Ruby Lee. Re-examining probabilistic versus deterministic key management. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 2586–2590. IEEE, 2007.

[51] Wenliang Du, Jing Deng, Yunghsiang S Han, Pramod K Varshney, Jonathan Katz, and Aram Khalili. A pairwise key predistribution scheme for wireless sensor networks. *ACM Transactions on Information and System Security (TISSEC)*, 8(2):228–258, 2005.

[52] Seyit A Camtepe and Bülent Yener. Combinatorial design of key distribution mechanisms for wireless sensor networks. In *Computer Security–ESORICS 2004*, pages 293–308. Springer, 2004.