Key Establishment for Wireless Sensor Networks Using Third Parties

by

Saleh Almowuena
B.Eng, King Saud University, 2006

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Master of Applied Science

in the Department of Electrical and Computer Engineering

Supervisory Committee

Key Establishment for Wireless Sensor Networks Using Third Parties

by

Saleh Almowuena
B.Eng, King Saud University, 2006

Supervisory Committee

Dr. T. Aaron Gulliver

Department of Electrical and Computer Engineering

Supervisor

Dr. Xiaodai Dong

Department of Electrical and Computer Engineering

Departmental Member

Abstract

Wireless sensor networks are employed in a wide range of applications including disaster relief operations, forest-fire detection, battlefield surveillance, pollution measurement, and healthcare applications. Because of the characteristics of these applications, a wireless sensor network is more vulnerable to security threats than traditional networks. In order to protect the sensor network from outside attacks, it is necessary to implement a cryptographic mechanism that can achieve three major security objectives: confidentiality, integrity and authentication. Even though the topic of cryptography has been well studied for traditional networks, many conventional cryptographic approaches cannot easily be applied to sensor networks. To illustrate, public key-based schemes and even some symmetric key methods are complex with regards to computations, memory, communication, and packet size requirements. On the other hand, sensor networks suffer from severe constraints on their available resources as a result of the necessity to increase the lifetime of the complete network, minimize the physical size of the sensor nodes, and reduce the cost of sensor nodes. Consequently, it is important to propose cryptographic solutions designed specifically for wireless sensor networks.

A fundamental element in an effective cryptographic system is how sensor nodes are equipped with the cryptographic keys needed to create secure radio connections with their local neighbours. This thesis contributes to the challenging field of key establishment by introducing three key agreement schemes whose memory, processing, and communication requirements are low. These methods utilize the concept of third parties, and sometimes also deployment knowledge, to reduce the cryptographic burden of public-key based schemes and the key management overhead of symmetric key approaches. The proposed methods employ just a few

simple hash operations in the sensor nodes. Furthermore, additional nodes called third parties are deployed to assist sensor nodes in the key establishment phase. Our key agreement schemes have many advantages over existing approaches. For instance, a sensor node in these schemes needs to make just a few local contacts to establish a secure radio connection with its neighbours with very high probability. In addition, the majority of sensor nodes must store only a small number of secret keys in their memory. These methods also employ an authentication mechanism to prevent impersonation attacks.

Table of Contents

List of Tables

List of Figures

Acknowledgments

First, I am extremely grateful to God for his mercy and divine guidance, without which this thesis would not be possible. It has been a great honour for me to be supervised by Professor Aaron Gulliver. His wide knowledge, enthusiasm in research, and pleasant personality has been a continued source of inspiration. My deepest thanks go to Dr. Gulliver for his insights and valuable feedback that contributed significantly to this thesis. Also, sincere appreciation is expressed to the Ministry of Higher Education in Saudi Arabia and the Saudi Arabian Cultural Bureau in Canada for their financial support during my graduate studies at the University of Victoria.

Dedication

I dedicate this thesis with heartfelt gratitude to the special people in my life: My two wonderful parents, whose encouragement and support are key factors in all my achievements; My beloved wife, Amani, whom I am forever indebted to for her continuous love and patience; and my close siblings, Nouf and Hamed, who were present in my heart during the composition of these pages. I also wish to dedicate this thesis to my friend, Rayyan, who recently passed away. Although he is not here with me, I will always feel his care and support pushing me to achieve my dreams.

Chapter 1

Background and Motivation

1.1 Introduction

During recent decades, the evolution of digital integrated circuits has made a dramatic development in various fields. For instance, the small size of these components accompanied with their reasonable cost has improved the application of physical phenomena detection. In order to obtain the temperature of a certain spot or notice an unauthorized intruder in a secure area, a large number of sensor units can be randomly deployed in the field of interest. Each node has a wireless transceiver and a computational processing unit. Once these sensor nodes are deployed, they construct a self-organizing wireless network in which each entity aims at observing events and sending information to a central node (typically named either the sink or base station). This type of configuration is often referred to as a wireless sensor network [1-3].

A typical topology of a wireless sensor network is shown in Figure 1, in which two kinds of devices with different capabilities are deployed arbitrarily in the region. In such a situation, sensor nodes are usually deployed in significant quantities (thousands to millions of units), in order to provide accurate zone coverage. For this reason, it is important to design them in an economical manner so that their manufacturing cost and physical size are small. However, reducing both parameters will eventually cause the nodes to have restricted bandwidth, a small memory unit, a limited power supply, and reduced computational capabilities. For example, a research group at the University of California, Berkeley succeeded in developing a reasonably priced sensor node named *Mica mote* whose physical size is only a few cubic inches but with

severe hardware resource constraints. To illustrate, the *Mica mote* [4] is equipped with a 4MHz 8-bit Atmel ATmega103 microcontroller, a 916MHz 40Kbps low-power radio transceiver covering up to few dozen meters, and two main memory units: 128 KB of RAM to store programming instruction and 512 KB of flash memory to store observation and other data. These nodes operate using just two AA batteries which provide around 2800 milliamps-hours at 3 volts. As a consequence, sensor nodes will function for approximately sixteen and half days in constant operation mode at 8 milliamps. This is possible because the Atmel ATmega103 processor consumes almost 5.5 milliamps at 3 volt in active mode and two orders of magnitude less power in sleep mode. Moreover, the *Mica mote* radio transceiver utilizes up to 12 milliamps, 4.7 milliamps, and 5 microamps in transmitting, receiving, and sleep mode, respectively [5].

Sensor nodes are equipped with a sensing module appropriate for the environment in which they are deployed. The parameters being monitored range from temperature, humidity, pressure, and pollution to motion, vibration and sound. Normally, sensor nodes continuously check the status of the target field to observe any changes in the environment. They send alert messages to report abnormal activities after processing the information and also respond to base station requests regarding the current situation of the area of interest. Of course, a significant number of sensor nodes will not be able to reach base stations directly because of their limited transmission capabilities, thus requiring relaying through intermediate nodes. As seen in Figure 1, wireless sensor networks also consist of other devices in addition to sensor nodes. Base stations are often control terminals which are many orders of magnitude more powerful compared to sensor nodes. Hence, they are greater than sensor nodes in their processing capabilities, memory units, radio transmission, and power supplies [1].

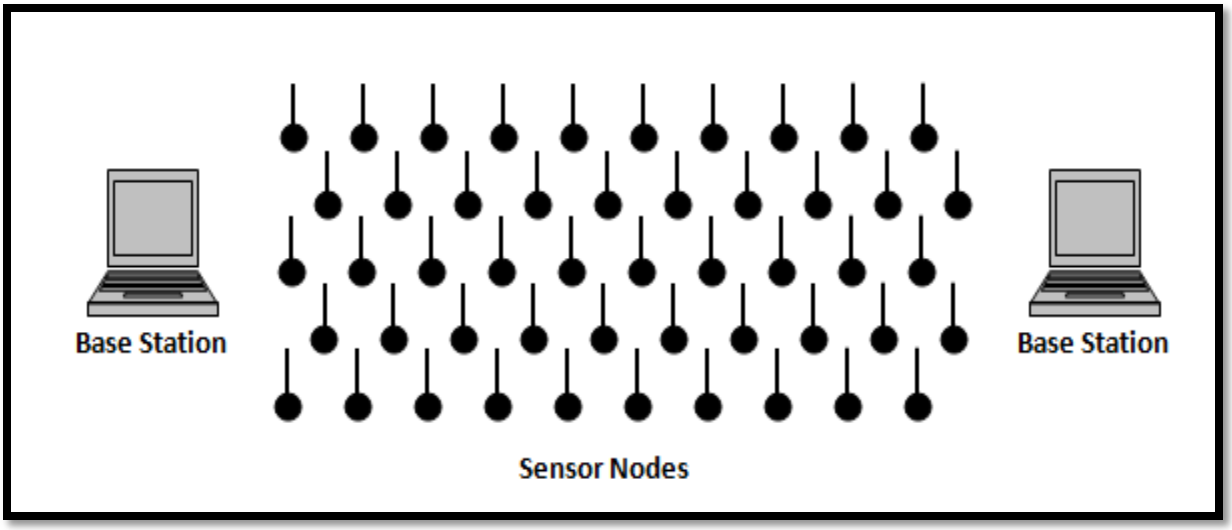

Figure 1: An example of a wireless sensor network.

Wireless sensor networks have received a lot of attention due to their inexpensive cost, high density, distributed nature, and robustness against a single point of failure. They are used in a wide range of applications including habitat monitoring, disaster relief operations, forest-fire detection, battlefield surveillance, pollution measurement, healthcare applications, and inventory control management. For instance, sensor nodes can be deployed randomly in a dangerous area instead of exposing emergency teams to hazardous chemicals or lethal gases. Sensor nodes in such situations will estimate the risk level by identifying chemical substances and measuring their concentration. Based on the obtained data, commanders can reliably decide which rescue and evacuation procedures are appropriate under the current circumstances [6].

Because of the characteristics of their applications as well as the broadcast nature of the radio transmissions, wireless sensor networks are more vulnerable to security threats than traditional wireless networks. Possible attacks range from passive eavesdropping to active impersonation, message replay and packet distortion. Furthermore, nodes can be physically compromised and captured since they are distributed in the field of interest in the absence of security guards and the lack of tamper-resistant modules. Therefore, it is important to develop

security schemes suitable for sensor networks as conventional security techniques are typically complex in terms of memory, computational, and communication requirements. Even though wireless sensor networks are considered as a special category of ad-hoc networks [3], their protocols are not suitable for five reasons:

- *Increased Density*: Most wireless sensor applications require thousands to millions of sensor nodes so that an acceptable detection accuracy of the physical phenomenon is achieved. This immense number of nodes creates problems in channel allocation procedures, collision avoidance mechanisms, and key management schemes.
- *Dynamic Topology:* Wireless sensor networks often operate in outdoor environments, and sometimes in hazardous regions. Thus, it can be anticipated that the node failure rate is high, leading to frequent changes in the network topology.
- *Limited Resources:* Different from traditional ad-hoc networks, nodes in wireless sensor networks are usually inaccessible, especially after the deployment phase. For this reason, it will be difficult to perform maintenance such as replacing or recharging the power supplies in the network. This results in four distinct constraints: a limited power supply, a restricted transmission range, a small amount of memory, and reduced computational capability.
- *Absence of Addressing:* Because of the enormous number of sensor nodes in the network, it is impractical to assign global identification or implement an addressing scheme. The lack of this important information presents many challenges to packet routing and node-to-node communication.
- *Nature of Communication:* In wireless sensor networks, nodes usually try to send their local observations to a central point (sink). To accomplish this objective, each node will

broadcast its packets and intermediate nodes may deliver these packets to the sink. The nature of communications is thus many-to-one.

Since conventional security techniques are complex, they are not suitable for use in wireless sensor networks. This thesis contributes to this challenging field by introducing three key agreement schemes whose memory, processing, and communication requirements are low. These methods utilize the concept of third parties to reduce the cryptographic burden of public-key based schemes and the key management overhead of symmetric key approaches. The proposed methods replace high cost public-key operations at the sensor nodes with a few and simple hash operations. Furthermore, additional nodes called third parties are deployed to help key establishment between sensor nodes. All three approaches have many advantages over existing techniques. For instance, a sensor node in these schemes needs to make just a few local contacts to establish a pair-wise connection with its neighbours with very high probability. In addition, the majority of sensor nodes must store only a small number of keys in their memory. Besides these benefits, the methods introduced here include an authentication mechanism to prevent impersonation attacks.

The rest of the thesis is organized as follow. Before introducing the three proposed key agreement schemes, this chapter presents a concise background of security issues in sensor networks and other important considerations including: key challenges, network architectures, adversary models, trust assumptions, security objectives, significant attacks, and defensive measures. From our perspective, this helps not only in understanding the security problems in wireless sensor networks, but also in defining a basis for evaluating the performance of the proposed methods. Chapter 2 provides a survey of current key agreement techniques in the literature. Chapter 3 and Chapter 4 describe the proposed schemes and present an analysis of

their performance with respect to: local connectivity, resilience to node capture, memory usage, and communication as well as computational overhead. To conclude, Chapter 5 provides some promising research directions for the future.

1.2 Sensor Network Obstacles

As a result of their dynamic nature and potentially hostile environments, wireless sensor networks encounter a variety of challenges that complicate the process of designing security protocols. For example, sensor nodes must be inexpensive in cost, reliable in performance, accurate in detection, and efficient in power consumption [7]. Furthermore, the network level should increase network lifetime, tolerate node failure, minimize the need for maintenance, provide scalable protocols, facilitate network discovery, meet predefined services qualities, and secure the entire network.

1.2.1 Lifetime

One of the most important challenges in wireless sensor networks is to increase the lifetime of the entire network [7]. Sensor nodes will sense a physical phenomenon, perform processing operations on detected events, store data, and transmit observations to a central point. These actions obviously consume power resources, thereby making power consumption a primary consideration. For instance, the *Mica mote* will exhaust its entire power supply during the first two weeks of deployment if it continuously operates at full power. To increase node lifetime to years, it is essential to run the sensor node on a duty cycle of less than one percent [5]. Besides, many experiments have shown that transmitting a single bit uses the same power resources as executing 800 to 1000 computational instructions [8-9]. Under such circumstances, much research has been conducted to decrease the amount of energy used in data transmission

and communication. For example, the Medium Access Control (MAC) layer has been modified to minimize channel listening without increasing collisions and packet retransmissions. Moreover, the network layer has been modified to control the packet reception, packet forwarding, and routing table update exchanges [7]. Thus, developing an efficient key management scheme requires the economical consumption of available power resources.

1.2.2 Fault Tolerance

Some sensor nodes may die due to unexpected troubles in their physical components such as a power supply shortage, obstacles in the terrain where they are deployed, the presence of noise, or the occurrence of jamming [10]. As mentioned earlier, wireless sensor networks operate most of times in outdoor or perilous environments so that crucial events can be observed and detected within an acceptable response time. Because of this criticality, the area of interest should be covered with a satisfactory amount of nodes every moment even though a set of sensor nodes become unreachable for whatever reason. In other words, any failure of a sensor node must encourage the network to adapt itself with sudden changes in topology, thereby not affecting the performance and connectivity of the entire network. [11] determines the probability for a node to remain in operation in the time period $[0, t]$ based on the Poisson distribution

$$R_k(t) = e^{-\lambda_k t} \qquad (1)$$

where λ_k represents the failure rate of node k.

1.2.3 Maintenance

As we know, replacing or recharging the power supplies of sensor nodes as well as repairing computation and transmission units is not possible in wireless sensor networks. Therefore, protocols must be designed to optimize use of available hardware in order to increase the lifetime of the network. On the other hand, the nature of sensor networks requires software

maintenance. For example, sensor nodes perform their tasks with the aid of a tiny operating system and a set of predefined programs. In the case when an administrator adds or removes a certain sensor node task, they have to change or update the programs running on the nodes by redistributing new software. In this scenario, wireless sensor networks should prevent the loss of update packets and authenticate the identity of the originators. Furthermore, sensor nodes should be able to self-configure themselves with the latest tasks. Another maintenance function is network synchronization since timing plays an important role in many network operations such as data aggregation and key management [7].

1.2.4 Scalability

Many wireless sensor networks take advantages of an enormous number of sensor nodes so that observation objectives are achieved. After the deployment phase, it is also possible to find situations where supplementary nodes are deployed in a random manner to improve network performance. As a consequence, protocols in wireless sensor networks must be designed with an ability to work in environments in which the number of sensor nodes is large. A further challenge for sensor network protocols is the region density, that is, the number of sensor nodes in a specific area. As a matter of fact, the density of a region will have a significant impact on various protocols and techniques [10]. To illustrate, exploiting flooding mechanisms to broadcast certain information in a network whose density is low can be an economic option, but it will consume a lot of resources in sensor networks with high density regions. The density of an area is given by [12]

$$\mu(R) = \frac{N\pi R^2}{A} \qquad (2)$$

where $\mu(R)$ represents the number of sensor nodes within a local neighbourhood, R refers to the radio transmission range, and N indicates the number of nodes deployed in a region of size A.

1.2.5 Network Discovery

In traditional networks, the infrastructure is frequently predefined in earlier phases, specifically before the deployment stage. Having such knowledge about the network topology will facilitate the process of packets routing as well as data gathering and then enabling the entire network to utilize its available resources in an efficient way. On the contrary, nodes in wireless sensor networks are always deployed randomly in the field of interest, thereby causing an arbitrary network topology. To overcome this, sensor nodes should be designed to have a self-ability to obtain information about their local neighbours before initiating data interactions [13]. Confining the knowledge of sensor nodes to the topology of their local regions will be adequate since sensor nodes only cooperate with the nodes within their transmission coverage. In some configurations, sensor nodes also need to be acquainted with their location, so such challenges must be encountered, especially if the cost of supporting global position systems is expensive.

1.2.6 Quality of Services (QoS)

Most applications aim at achieving a certain quality of service such as: maximizing the throughput of the entire network, minimizing the propagation latency of the response packet, and increasing the accuracy of event detection [14]. Indeed, the nature of wireless sensor networks presents a complicated challenge in how to find the optimal trade-off between these metrics. For instance, exploiting the redundancy in observation data will enhance the reliability and accuracy of event detection, but it will remarkably consume the network resources including throughput and energy. In contrast, requiring sensor nodes to analyze their collected data and then perform simple compression and aggregation operations will maximize the utilization of resources, but it may create a significant latency in the response time and affect the level of reliability of the

results obtained [13]. Therefore, protocols and algorithms in wireless sensor networks have to consider the preference of the applications using the sensor data.

1.2.7 Security

Another important factor in a wireless sensor network is to make it robust against threats affecting performance. This can be done by providing a set of security countermeasures such as: ensuring the data confidentiality to avoid any disclosure of sensitive information to unauthorized users; satisfying the data integrity to prevent not permitted alteration in transmitted packets; applying authentication procedures to verify the identities of both senders and receivers; and implementing a strong denial of service protection to guarantee absolute accessibility of network resources during the network lifetime. As a result of resources constraints, applying these necessary operations is a challenge [7].

1.3 Essential Considerations

Before discussing the security problems in wireless sensor networks, there are three important factors that should be considered: network architectures, adversary models, and trust assumptions. In the next subsections, we will outline two distinct architectures for sensor networks, suggest three different classes of adversary models, and point out a number of assumptions about the trust model in sensor networks.

1.3.1 Network Architectures

To increase the network lifetime and minimize end-to-end propagation delay, several applications in wireless sensor networks employ a hierarchical architecture. In this case, the sensor nodes are divided into clusters. In each cluster, a sensor node with superior capabilities is assigned as the central header or commander. The sensor nodes in a cluster forward their data to

the local commander. Commonly, these central headers are responsible for eliminating data redundancy, performing data aggregation, and then establishing a communication channel to a base station as shown in Figure 2. In some cases, this model is referred to by heterogeneous networks due to the reality that devices in these configurations show a discrepancy with regard to their resources including memory storage, computational processing units, transmission coverage, and power supplies. On the other hand, a number of sensor applications prefer to deploy their nodes in a distributed manner in which sensor nodes are arbitrarily scattered in the field of interest without any restriction to a predefined or fixed infrastructure as shown in Figure 1. In such scenarios, nodes will perform identical roles: discovering nodes in their local radio range, checking the status of the targeted region, collaborating with their neighbours to forward and aggregate obtained observations to base stations, which are more powerful than standard nodes. This model is called homogeneous since the sensor nodes are similar in their memory, computation, and communication capabilities. Note that the data flow in both of hierarchical and distributed models can be classified into three categories: pair-wise in which messages are unicast to a particular entity, group-wise in which packets are multicast within a single cluster, and network-wise in which data is broadcast throughout the network [15-16].

1.3.2 Adversary Models

In wireless sensor networks, security attacks can be classified based on three parameters: the identity of the participants and their roles in the system, the motivation for the attacks, and the capabilities of the tools utilized to launch these attacks [17-18]:

- *Outsider versus Insider Attacks*: outsider attacks refer to security threats launched by external adversaries or sensor nodes that do not belong to the wireless sensor network. An insider attack occurs when a legitimate sensor node behaves in an undesirable manner

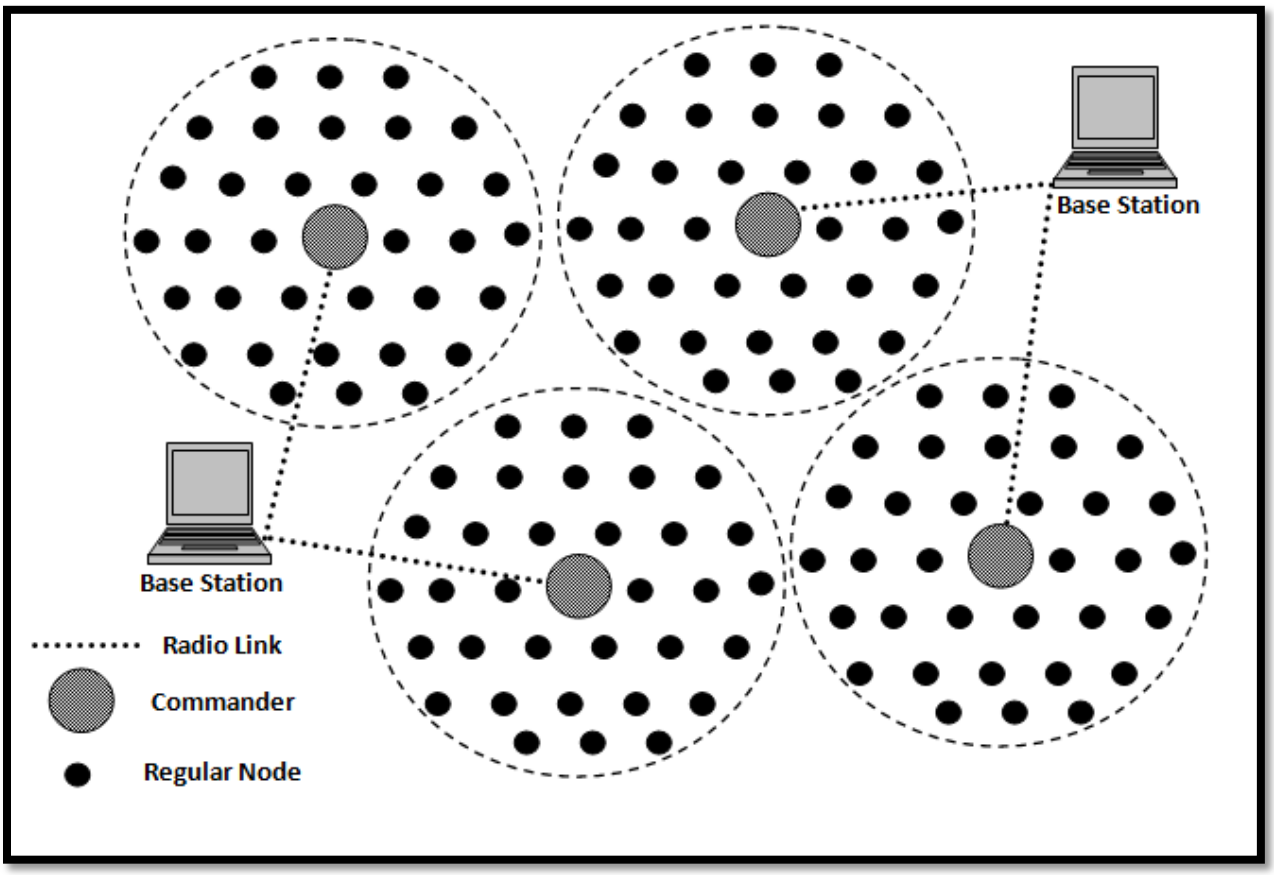

Figure 2: An example of a hierarchical wireless sensor network.

by accident or carries out unauthorized activities on purpose after being compromised. Clearly, the latter scenario is more difficult given that defense countermeasures, such as encryption and authentication, may not be able to overcome the security attack caused by a compromised node. This is due to the fact that capturing a sensor node reveals its key materials and programming instructions, thereby enabling an adversary to be treated as a legitimate entity.

- *Passive versus Active Attacks*: passive attacks indicate those situations where adversaries are able to monitor messages passing through radio channels. In those cases, attackers are trying to recognize information exchanged among entities and then obtain data observations gathered by sensor nodes. Conversely, active attacks specify those situations where adversaries are able to alter, modify, delete, or spoof the content of packets

originated from a genuine user and intended for other authorized entities. The danger here is such modifications can be directed at critical packets. For instance, an attacker can focus on altering routing information so that routing tables become inconsistent, thereby affecting the performance of the entire network. Another example is when an adversary generates fake messages containing wrong observations in order to mislead base stations and influence them to react incorrectly.

- *Node versus Laptop Attacks*: it is noteworthy to distinguish between security threats based on the categories of devices utilized to launch these attacks. To clarify, some adversaries exploit nodes with capabilities matching sensor nodes in the network. Because of the severe constraints in such resources, node-class attacks can only impact local vicinities, leading adversaries to only monitor and manipulate messages exchanged in their surrounding area. To extend their effect on the whole network, attackers can employ more powerful machines such as laptops and notebooks. With such devices, adversaries are able to overhear more packets, deceive a large number of sensor nodes, and even perform a jamming attack on the entire network. Compared to regular sensor nodes, laptop-class devices do not suffer from restrictions in power supplies, limitations in memory usages, or constraints in computational capabilities. Moreover, these machines generally have a wider transmission range, making attacks from this category further complicated.

1.3.3 Trust Assumptions

Throughout this thesis, several assumptions are considered in both the analysis and discussions. At this point, we will try to highlight the major assumptions, on the basis that other complementary assumptions will be mentioned in their relevant context:

- Both physical and data-link layers are vulnerable to direct attacks. Hence, wireless channels are insecure and susceptible to packet eavesdropping, data injection, message modification, and replay attacks.
- Sensor nodes are not supported with tamper-resistant modules since such components are either simple but insecure or robust but costly. As a consequence, an adversary can compromise a sensor node and extract its key materials, observation data, and software.
- Base stations are trustworthy devices, whereas sensor nodes are not guaranteed to behave reliably.
- An attacker is capable of employing any of the adversary models discussed earlier. As an example, he can deploy malicious nodes with resources identical to those found on the sensor nodes or launch attacks taking advantage of powerful laptops. Furthermore, an attacker can compromise a number of regular nodes and manipulate them to start a collusion attack against the network.
- An adversary targets sensor nodes randomly without prior knowledge of the keys carried on the nodes. Nevertheless, he may be aware of the security mechanisms implemented in the wireless sensor network.

1.4 Security Objectives

Wireless sensor networks are extremely vulnerable to various types of threats due to the strict limitations in their resources, the broadcast nature of their transmission channels, and the uncontrolled characteristics of environments in which the nodes are scattered. Accordingly, it is necessary to enhance the protection of data transmission and processing within these networks by ensuring the achievement of certain objectives [24]. Generally, these goals involve: the

confidentiality of messages, the integrity of data, the authentication of entities, the freshness of packets, the availability of resources, and requirements specific to wireless sensor networks.

1.4.1 Confidentiality

Sensor nodes in various applications, particularly those implemented in the military and healthcare sectors, monitor critical operations and send crucial reports to the trusted base stations. Therefore, nodes in wireless sensor networks should prevent the disclosure of their secret information and observations by unauthorized entities. Normally, sensor nodes accomplish this objective by encrypting packets with keys distributed prior to deployment and establishing a secure channel among legitimate parties.

1.4.2 Integrity

Applying the concept of confidentiality does not ensure that received packets are exactly what the senders transmitted, especially in multi-hop networks where intermediate nodes can easily insert, delete, and substitute content into messages during their journey to destinations. Sometimes, these modifications are caused by errors as a result of deployment in a harsh environment, which may cause a noticeable increase in data loss [24]. Therefore, nodes in wireless sensor networks should make certain that incoming packets have not been altered either by accident or by malicious acts.

1.4.3 Authentication

Adversaries not only have the capability to monitor and manipulate messages exchanged between sensor nodes but they are also able to masquerade the identities of trusted units. In the latter scenario, intruders will impersonate the identification of a genuine device, construct new packets with the stolen identity, inject packets into the transmission medium, and then deceive other nodes about the accurate origin of these messages [25]. To overcome this problem,

recipients are required to recognize the correct source of packets by employing message authentication codes. In this case, both sender and receiver share a secret key in advance, so they can add a short signature to every data interaction in order to explicitly confirm their identities.

1.4.4 Freshness

A straightforward form of identity impersonation would be the case in which attackers exploit the availability of old packets already sent by legitimate users [25]. Using these old packets, an adversary can deceive sensor nodes. To illustrate, base stations regularly broadcast inquiries about the current status of the physical phenomena. Sometimes, they also update the software running on sensor nodes to add or remove certain tasks to the system. If an intruder captures these requests, he may use a replay attack to enquire about recent observations or to forward old software, leading to unauthorized interference in network operations. A useful countermeasure for such attacks would be to add a nonce (i.e. a parameter which varies with time) to packets so that data freshness is guaranteed.

1.4.5 Availability

Wireless sensor networks play a significant role in a wide range of applications including habitat monitoring, disaster relief operations, forest-fire detection, battlefield surveillance, pollution measurement, healthcare applications, and inventory control management. These critical applications desire continuous availability of the sensor nodes for constant surveillance of the sensitive terrains [25]. As a consequence, to increase their lifetime nodes should avoid using security mechanisms whose memory, computation, and communication requirements are complex. Moreover, they should employ protocols such that the probability of a single point of failure is minimized.

1.4.6 Specific Requirements

The previous objectives are typically required to secure both traditional and sensor networks. However for wireless sensor networks, it is important to adjust the level of security according to the remaining power levels [16]. It is also essential to guarantee a minimum set of security services in the presence of low energy resources, the existence of attacks, or even the occurrence of increased node failure rates [15]. A further goal is providing both forward and backward secrecy. Forward secrecy ensures sensor nodes which were excluded from the network for whatever reason are not allowed to read future messages. Quite the opposite, backward secrecy means nodes which just joined the network are not able to decrypt packets exchanged prior to their enrolment [17].

1.5 Typical Sensor Network Attacks

Because of the close resemblance between their structures, significant attacks in wireless sensor networks are similar to those in ad-hoc networks. As with wired or ad-hoc networks, security attacks in sensor networks can be classified into two categories: attacks aimed at breaking security mechanisms implemented in the application layer and attacks focusing on disturbing the performance of routing algorithms utilized in the network layer. The first class can be defended against by choosing measures appropriate for the characteristics of the required applications. For example, the strength of the security protocols needed in battlefield surveillance and disaster relief operations will be completely different from those applied to applications designed to inquire about the current traffic status or to obtain statistics about the number of visitors in a tourist attraction. Due to the simplicity of routing algorithms in wireless sensor networks, it may be easy to modify and spoof messages exchanged among sensor nodes, thereby

making routing tables inconsistent and causing packets to be routed to unintended destinations. Furthermore, an adversary may inject fake packets and send unnecessary requests in order to consume communication, computation, and memory resources. Important attacks at the networks layer can be categorized in one of the following groups: compromising routing information, selective forwarding, sinkhole attacks, sybil attacks, wormholes, hello flood attacks, and acknowledgement spoofing.

1.5.1 Compromising Routing Information

Section 1.3.3 emphasized the vulnerability of both the physical and data-link layers in wireless sensor network to active and passive attacks. Under this circumstance, intruders can monitor messages transmitted through the wireless channels and have the ability to interact with legitimate entities in the network. For instance, they may modify, discard, replay, or spoof routing information exchanged between sensor nodes causing a negative impact on network performance and resource availability. Such disruptions include but are not limited to increasing the end-to-end latency, attracting or discouraging network traffic from certain units, dividing the network into various segments, generating fake error messages, and providing misleading information to base stations [5, 19-20]. To overcome attacks in this category, it is recommended to use message authentication codes in order to recognize the identity of participants and to add a time nonce to packets in order to prevent replaying previous messages.

1.5.2 Selective Forwarding

A feature of wireless sensor networks is the limited radio transmission range which requires sensor nodes to rely on intermediate nodes in order to deliver observations to the base stations. However, implementing the concept of multi-hop networks may introduce risks, especially when intermediate nodes are not guaranteed to be trustworthy. A possible threat can

be demonstrated in selective forwarding, that is, scenarios where malicious nodes forward packets originating from particular sensor nodes and drop other messages. In some cases, these malicious components will try to act as a central point in the network and drops all incoming data, forming what is called a black hole [5, 19-20]. A simple solution for attacks in this category is to enforce legitimate nodes to duplicate their messages and then forward these packets through two distinct routes with the hope that at least one copy will survive. Obviously, the significant drawback of this approach is the consumption of node resources and the transmission of redundant packets.

1.5.3 Sinkhole

As mentioned earlier, adversaries are assumed to have the ability to exploit powerful devices to launch their attacks. Based on this, intruders can choose nodes with attractive specifications, such as substantial transmission power and energy resources, and then announce the possession of these alluring parameters to draw the attention of sensor nodes in the whole network. A significant quantity of sensor nodes may be deceived about the accurate location of adversaries, so they will try to forward their observation packets through these malicious nodes in spite of the fact that these powerful nodes are not within their local neighbourhoods. Unfortunately, victims may also unintentionally mislead and encourage other nodes in their surrounding area to change their routing tables so that malicious nodes become preferred gateways to base stations. As a result, adversaries receive packets from almost every node in the network, facilitating a selective forwarding attack [21]. Sometimes, attackers will not be able to acquire powerful nodes, yet they can construct fake announcements about the availability of high-quality routes simply by manipulating or copying real advertisements. This scenario should be easily detected by measuring the propagation latency of response messages.

1.5.4 Sybil

Since wireless sensor networks are typically dense, there is usually redundancy in the data observations obtained by sensor nodes. Several protocols have been proposed to exploit this redundancy. However, an adversary can disrupt the functioning of these protocols by launching a sybil attack in which they use multiple identities with a single node [22]. This can easily be done in wireless sensor networks due to the absence of addressing schemes. Another example of a sybil attack would be the situation when base stations collect a distinct set of measurements with varying values. Normally, a conflict in data observations is reasonable because sensor nodes have different locations relative to the targeted event. Hence, base stations will look at each incoming observation as a vote and make decisions based on the majority of received votes. Sybil attacks can use a single sensor node to forward false data observations signed with multiple identities to mislead the base stations and enforce them on performing unnecessary reactions.

1.5.5 Wormholes

Wormholes refer to those attacks in which an adversary receives a message from a certain portion of the sensor network and then tunnel this message through a low latency channel to another segment of the network [23]. Two probable settings for the attacks in this category are illustrated in Figure 3: 1) an intruder utilizes a single node to get packets from a sensor node and then forward those packets to a non-neighbouring node; and 2) an adversary exploits two powerful devices to connect distant clusters. As shown in Figure 3, an attacker employs a device in the surrounding area of a base station, deploys another laptop in a remote segment of the sensor network, and establishes a communication channel between the two sections. Under this circumstance, the adversary can announce in the distant portion about the availability of a high-quality route to a base station. Naturally, this low latency link will draw the attention of nodes in

that region to take an advantage of this attractive route, thereby enabling the intruder to implement both sinkhole and selective forwarding attacks.

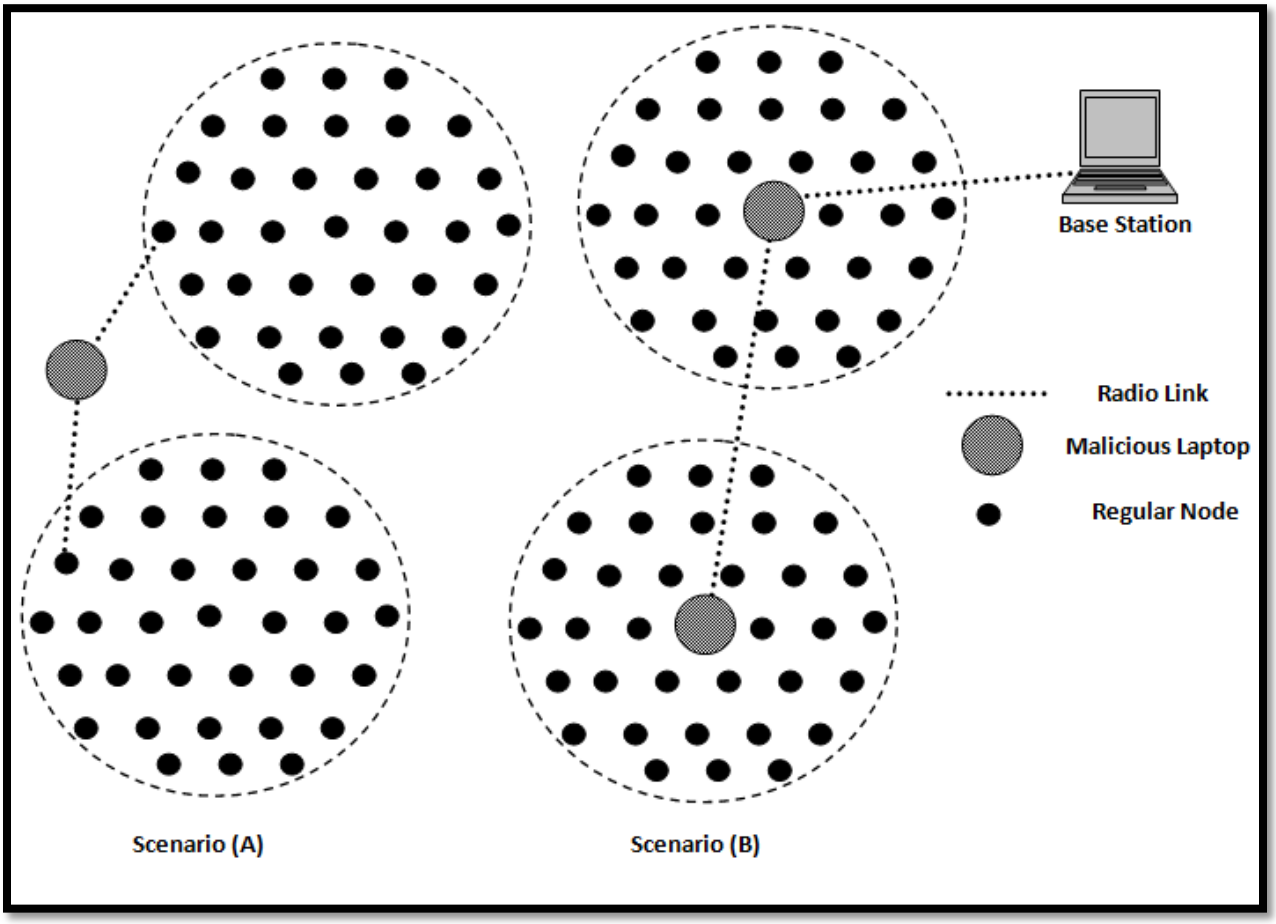

Figure 3: Two probable scenarios for wormholes attacks.

1.5.6 HELLO Flood Attacks

To minimize the consumption of power resources, wireless sensor networks use simple protocols. For instance, many algorithms utilize a naive mechanism to discover neighbours in a local cluster. For example, sensor nodes may use HELLO packets or what is called beaconing to advertise their existence in a neighbourhood. Since adversaries have the capability of exploiting devices with considerable transmission coverage, they can send HELLO packets to every node in the network. Innocently, a significant number of sensor nodes may consider these powerful

devices as neighbours and try to take an advantage of the high-quality routes even though in reality most of these malicious nodes are located out of their radio ranges [5].

1.5.7 Acknowledgement Spoofing

Similar to conventional networks, several protocols in wireless sensor networks require data-link layer acknowledgments either explicitly or implicitly. Because of the broadcast nature of the network, an adversary may capture an acknowledgement and manipulate its content to deceive other entities in the network [5]. For example, in a HELLO flood attack, a sensor node can challenge the adversary to confirm its existence in the surrounding area. To accomplish such verification, the adversary can simply spoof an acknowledgment sent by one of the challenger's neighbours. Another attack in this class can be found in the process of updating routing tables. Each sensor node will periodically update its routing table in order to adapt changes in the network topology. When a neighbour becomes unreachable for whatever reason, the sensor node will eliminate its routes from the table's entries. However, intruders can make the latest routing tables inconsistent by taking the role of a dead node and sending its previous acknowledgement announcing that the dead node is still alive. Encouraging sensor nodes to send their packets to a dead node will cause forwarded message to be dropped, thus providing negative impact similar to that resulted from selective forwarding attacks.

1.6 Defensive Measures

To protect wireless sensor networks from outside attacks, it is necessary to implement a cryptographic mechanism which can achieve three major security objectives: confidentiality, integrity and authentication [26]. Even though the topic of cryptography has been studied well in most traditional networks, many conventional cryptographic approaches are not feasible to be

applied on wireless sensor networks. To illustrate, well-known public key schemes (and even some symmetric methods) are complex in their computation, memory, communication, and packet size requirements. On the other hand, wireless sensor networks suffer from severe constraints in their available resources in order to increase the lifetime of the complete network, minimize the physical size of nodes, and reduce the cost of sensor nodes. Consequently, it is important to propose cryptographic solutions designed specifically for sensor networks.

A fundamental element in an effective cryptography will be how secret key is properly managed with regards to its three main components: distribution, renewing, and revocation. Typically, key distribution represents the process of equipping every node with the cryptographic keys needed to establish secure radio connections with its local neighbours. Once in a while, these secrets keys should be refreshed and renewed to boost the security of the entire network. Furthermore, key management methods require providing an appropriate mechanism to revoke the secret keys of compromised nodes and ensure of their inabilities in the future to decipher any sensitive message exchanged between the sensor nodes. This thesis focuses on the issue of key distribution in wireless sensor networks and aims at contributing in this challenging field by proposing three pair-wise key establishment schemes whose processing, communication, and memory requirements are low.

Chapter 2

Related Work

2.1 General Overview

A simple technique to enable sensor nodes establishing secure communication channels with their neighbours is to preload a single shared key into the memory space of each node in the network. Having the same secret key provides an efficient key agreement scheme with regards to both power consumption and memory usage. In order to secure their radio links, sensor nodes utilize only a single unit of their available storage resources and avoid performing any data interaction. However, this method offers weak resilience against node capture since compromising any sensor node will reveal its key material and then cause a major breach in the security of the entire network. To enhance the resilience against node capture, a pair-wise key can be assigned to every pair of nodes in the network. In this case, trusted base stations should randomly generate $\frac{N(N-1)}{2}$ secret keys and then supply each sensor node with $N-1$ of these keys. A sensor network may have thousands to millions of nodes in the field of interest, so it is not practical to implement such an approach because of the limitation in memory resources.

Under the assumption that base stations are considered trustworthy entities in wireless sensor networks, it is easy to apply the concept of Key Distribution Centre (KDC) in which base stations are responsible for assisting sensor nodes in the process of key establishment. To do so, each sensor node in the network will share a unique symmetric key with a base station. Once neighbours in its transmission range are identified, the sensor node forwards a request to the base

station indicating a list of sensor nodes with which it intends to initiate secure radio connections. Responding to this request, the base station generates a set of pair-wise keys and then transmits them back to the sender. The main drawback here is the possibility of a single point of failure as well as the number of packet transmissions.

Recently, researchers have focused on the idea of key pre-distribution in which a large pool of symmetric keys is generated before deploying sensor nodes in the target terrain. Each sensor node is equipped with a ring of secret keys chosen randomly from a key-pool. Two sensor nodes will be capable of creating a secure radio connection if both share a common key. Usually, the key-rings in pre-distribution approaches are designed in such a way that nodes can succeed with a pre-determined probability in finding shared keys with their local neighbours. Regarding the performance of the methods in this category, it has been shown that the number of symmetric keys in both the key-ring and the key-pool significantly affect the network connectivity, the resilience against node capture, and the memory usage. For example, increasing the number of secret keys in the general key-pool will enhance the security of the entire network, but it may negatively impact the local connectivity of sensor nodes. To balance this trade-off between security and connectivity, efforts have been made to optimize traditional public key-based algorithms, such as elliptic curve cryptography, in order to make them suitable for wireless sensor networks. This chapter briefly reviews a number of schemes that have been proposed using either random pre-distribution or public keys.

2.2 Random Key Pre-distribution Schemes

The *Basic scheme* proposed by Eschenauer and Gligor in [27] can be considered the "fundamental" method in random key pre-distribution approaches. Simply, this scheme consists

of three phases: distributing secret shares, discovering local neighbours, and establishing secure radio connections. Prior to the deployment of a sensor network, a trusted base station preloads a ring of keys into the memory of each node in the network. Each key ring, or what is sometimes called key-chain in the literature, consists of k secret keys chosen arbitrarily from a large pool of P keys, which is generated off-line in advance by trusted authorities. After scattering sensor nodes in the field of interest, these nodes perform a simple discovery process in order to find neighbours with whom they share at least a single secret key in common. Once the discovery stage is completed, sensor nodes in a given area can exploit their shared keys to establish secure communication channels. However, it is possible to encounter situations in which a shared key does not exist between a pair of sensor nodes since this scheme is designed to achieve a pre-defined probability of local connectivity. In other words, the probability that two sensor nodes in a neighbourhood can interact using a pair-wise key is not 100% due to the randomness in the selection of key rings. In such scenarios, senor nodes may take advantage of intermediate nodes to exchange their secret keys so that a direct and secure link can be initiated.

Many key agreement methods have been introduced to improve the reliability, efficiency, and security of the previous solution. For instance, Chan et al. in [28] presented a *q-composite random key pre-distribution scheme* in which each pair of sensor nodes requires $q > 1$ shared secret keys to establish a secure communication channel, instead of using just a single key as explained in the *basic scheme*. Obviously, the network resilience against node capture in this case is greatly enhanced, especially when q is large. On the other hand, this improvement here will cause either a reduction in the local connectivity of the sensor network or a significant increase in the memory usage of sensor nodes.

Hwang and Kim in [29] proposed a *transmission range adjustment scheme* in which each sensor node has the capability to enlarge its radio transmission range during the phase of discovering local neighbours and then reduce it back to a predefined range as soon as the shared keys with neighbours have been identified. To apply such a feature, the authors slightly modify the *basic scheme* in [27] as follows. Once sensor nodes are spread in the target terrain, they temporarily adjust their radio transmission range to a predefined radius r_{max} in order to obtain both a desirable number of neighbours and a required number of secure links. At this point, adjacent nodes sharing a secret key in common will be able to create secure radio connections. Completing the key establishment phase, sensor nodes will readjust their transmission power back to the predefined range, thereby conserving power in the entire network. In the case when the neighbouring nodes i and j in a local region do not share a common key, one of the two will take the responsibility of generating a new secret key and forward this key to the other partner via intermediate nodes with which both sensor nodes have already established secure links.

As mentioned in Chapter 1, many experiments have shown that transmitting a single bit uses the same power resources as executing 800 to 1000 computational instructions [8, 9]. Under such circumstances, some researchers have tried to improve the *basic scheme* in order to decrease the amount of energy used in data transmission and communication. For example, Hwang et al. introduced in [30] a *cluster key grouping scheme* whose main objective is reducing the number of packets exchanged in the neighbour discovering step. To accomplish this goal, the method divides the key-pool into M clusters in which each cluster consists of W secret keys. Prior to the deployment stage, sensor nodes are randomly equipped with C clusters. When nodes are distributed in the environment of interest, each sensor node is required to broadcast only the identities of its C clusters. These identities are implicitly designed to help nodes in determining

the identities of the W secret keys in each cluster. To clarify, the method here forwards just C identities when discovering neighbours rather than sending $k = C \times W$ identities as discussed in the *basic scheme*, thereby allowing a decrease in the power consumption of nodes. As with the previous approaches, if two sensor nodes share at least one common secret key, a secure communication channel can be established directly. Otherwise, they need to exploit intermediate nodes to create a secure radio link.

Another important attempt at eliminating the communication overhead caused by the neighbours discovering stage can be seen in [31]. This work exploits a random key pre-distribution mechanism similar to that found in the *basic scheme* and also takes advantage of the idea of threshold secret sharing presented by Shamir in [32]. Combining both concepts results in an approach with two distinct properties: 1) each pair of sensor nodes utilizes an exclusive pair-wise key in order to establish a secure radio channel, and 2) the discovery process to identify shared keys among sensor nodes in a surrounding area is not required. To illustrate, before the deployment stage, the trusted base stations arbitrarily generate a large number of secret keys associated with their unique identifiers. To preload a sensor node with its corresponding key-chain, a base station uses the node's identity as input to a pseudo-random number generator in order to get the key identifiers for the node. At this point, equipping sensor nodes with the same pseudo-random number generator can eliminate the neighbour discovery stage because each node will have access to the key identifiers of its neighbours to determine whether they share a secret key or not. To guarantee the exclusive usage of pair-wise keys among sensor nodes in the network, a node should share more than a secret key with each of its neighbours. In cases when a single key or less exists, they must rely on intermediate nodes to complete the key establishment process. After ensuring the existence of sufficient paths to its neighbours, the sensor node will

generate a random secret key S for each of its neighbours. After that, it will split each secret key S into multiple shares with the aid of a secret sharing mechanism and send these shares via separate logical paths to the corresponding neighbours. Upon receiving all these shares, the neighbouring node will use a reconstruction mechanism to combine incoming shares in order to obtain the original secret key S.

To conclude the discussion of random key pre-distribution approaches, we consider a noteworthy scheme introduced by Du et al. in [33]. To predict the location of a sensor node in a specific region with a certain probability, the authors in [33] presented a deployment knowledge model following the Gaussian probability density function. Then they used the proposed model with a random key pre-distribution method based on the *basic scheme* in [27]. To explain, a trusted base station in the pre-deployment phase divides the sensor nodes into $t \times n$ groups with the aim of positioning each group of nodes at a particular point in the field of interest. The desired location of a group of sensor nodes is referred to as the resident point. The base station also divides its own pool of secret keys into $t \times n$ sub-pools and allocates each group of sensor nodes one of the sub-pools. Usually, the key pool is divided such that the sub-pools of neighbouring groups share a set of secret keys in common. As a consequence, sensor nodes at the group boundaries may have the ability to establish a secure radio channel with other nodes in adjacent groups and thus avoid the problem of isolated groups. To distribute the secrets keys to sensor nodes in the network, the trusted base station will preload each node with keys randomly selected from its corresponding sub-pool. Once nodes are scattered in the target environment, sensor nodes may not be exactly located at their resident point, but they will be close to this position with a high probability. Indeed, the simulation results obtained in [33] show that using of deployment knowledge with the *basic scheme* improves the local connectivity of the entire

network, strengthens the resilience against node capture, and reduces the amount of memory required to store secret keys.

2.3 Pair-wise Key Pre-distribution Schemes

The majority of random key pre-distribution approaches discussed in the previous section do not assure exclusive use of secret keys. In other words, several communication channels may be secured using the same key. This vulnerability will negatively impact the resilience against node capture since compromising a sensor node will not only reveal its key materials but also facilitate the process of breaking the secrecy of other radio channels using the same key materials. To overcome this vulnerability, pair-wise key agreement methods assign a unique shared key for each pair of sensor nodes. Unlike conventional pair-wise schemes in which sensor nodes should be equipped with $N - 1$ secret keys, the authors in [28] takes an advantage of the reality that nodes in a wireless sensor network are not capable of interacting with nodes located beyond their radio transmission coverage. Hence, the key-chain of a sensor node can be limited to a number of pair-wise keys that is equivalent to the expected quantity of neighbours in its surrounding area, thereby minimizing the storage needed for the key establishment phase. To accomplish this objective, trusted base stations should assign each sensor node with a unique identity during the pre-deployment stage. For each sensor node y, base stations choose a set of nodes based on a random selection of the node identities. Then they generate a pair-wise key for every node in the resulting set, and preload these pair-wise keys with their corresponding node identity into the memory units of the sensor node y. Normally, the number of pair-wise keys stored in a sensor node will be chosen such that the probability of two nodes establishing a secure radio connection meets a desired percentage. Once sensor nodes are deployed in the target

region, each node will perform a key discovery operation in which its identity is broadcast within the local neighbourhood. Receiving these identities, neighbouring nodes will look up in their memory space to determine whether a pair-wise key exists or not. If there is a shared key between the two nodes, the destination node will response to the sender with a cryptographic handshake indicating the possibility of initiating a secure communication channel.

Since the random pair-wise key scheme [28] is not scalable to the addition of nodes in the network, Liu and Ning in [43] presented a location-based pair-wise key establishment approach in which a deployment knowledge model was exploited to improve scalability and enhance network connectivity. Prior to scattering sensor nodes in the field of interest, trusted base stations divide the two-dimensional field into small cells and locate every node at a particular resident point. Furthermore, base stations generate a distinct private key S_i for each sensor node i. To construct the key-chain belonging to a particular node, base stations begin by determining the c closest neighbours to this node. Next, pair-wise keys for these c neighbours are computed using a pseudo-random function as following

$$K_{i,j} = PRF(S_j | ID_i) \text{ where } j = 1, 2, \dots, c$$

Clearly, this method provides good scalability as new nodes can be equipped with the pair-wise keys of the c sensor nodes closest to their predicted location. Compared to the random pair-wise key scheme in [28], the solution here provides a similar resilience against node capture but offers better local connectivity, especially in cases where the distance between the deployment point and the resident point is small. Regarding memory usage, each sensor node must utilize $2c + 1$ of its memory units to facilitate the process of establishing secure radio channels.

2.4 Matrix-based Key Pre-distribution Schemes

Rolf Blom in [34] presents a technique that achieves local connectivity with all neighbours. Blom's method does not create a significant burden on the node resources including memory usage, computational processing, or power to exchange packets in the key agreement phase. Assuming N sensor nodes in the field of interest, the scheme in [34] generates over the finite field $GF(q)$ two different types of matrices: a public matrix G with dimensions $(\lambda + 1) \times N$ and a private matrix D with dimensions $(\lambda + 1) \times (\lambda + 1)$. To construct the pool of secret keys, trusted authorities multiply both matrices according to

$$K = (D \times G)^T \times G \qquad (3)$$

K is an $N \times N$ symmetric matrix from which the keys of all possible radio links in the sensor network can be obtained. Prior to the deployment stage, each sensor node is equipped with its own secret shares based on its unique identity as follows. Sensor node S_i is provided with the ith column of G as its public information in addition to the ith row of $(D \times G)^T$ as its private information. When sensor nodes are spread in the target terrain, they begin by discovering nodes in the surrounding area. To help in establishing a shared key, neighbours S_i and S_j exchange their public information, $column_i$ and $column_j$, respectively. Then each node can generate the same pair-wise key using its private row as follows

$$K_{ij} = row_i \times column_j \Leftrightarrow K_{ji} = row_j \times column_i \qquad (4)$$

In addition to guaranteeing complete connectivity for sensor networks, Blom's scheme was designed to ensure robust resilience against compromised nodes in cases when the number of captured nodes is smaller than the threshold λ. However, compromising more than λ sensor nodes will compromise the security of the other node links. To enhance the resistance of Blom's

method, Du, Deng, Han, and Varshney (DDHV) in [35] applied the concept of the *basic scheme* in [27] to the idea of the Blom's scheme in [34]. They employ a single public matrix G and a set of ω private matrices D in order to construct distinct spaces $\{(D_i, G)$ for $i = 1, \ldots, \omega\}$ of secret keys. Before distributing sensor nodes in the region, trusted base stations assign to each node a set of τ spaces randomly chosen from the ω general spaces, with τ in the range $(2 \leq \tau < \omega)$. Next, each sensor node is preloaded with its unique shares of private and public information obtained from its corresponding spaces $\{(D_i, G)$ for $i = 1, 2, \ldots, \tau\}$ using Blom's scheme. Hence, each node must store $\tau + 1$ vectors of size $\lambda + 1$ in its memory units. Moving to the shared key discovery step, a pair of adjacent sensor nodes will exchange a message indicating the identifiers of their τ spaces. After choosing a common space, each node will broadcast its corresponding public column to allow other nodes to generate a shared key. Like previous methods, a pair of nodes should depend on intermediate nodes in situations when no common space exists.

2.5 Polynomial-based Key Pre-distribution Schemes

Instead of using public and private matrices, Blundo et al. in [36] proposed a polynomial-based key pre-distribution scheme using a bivariate t-degree polynomial

$$f(x, y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j \qquad (5)$$

generated randomly over the finite field $GF(q)$. To facilitate the process of key agreement in a sensor network, this bivariate t-degree polynomial should possess two main properties: 1) it is a symmetric function in which the equation $f(x, y) = f(y, x)$ is satisfied, and 2) it is generated over a finite field $GF(q)$ in which the prime parameter q is large enough to accommodate a cryptographic key. In the pre-deployment phase, the trusted base stations evaluate (5) for each

sensor node in the network. To perform this operation, the base stations input the identity of the sensor node i into (5) so that the corresponding secret share $f(i, y)$ is generated. Once the sensor nodes have their polynomial share, every pair of nodes in the wireless sensor network will be able to agree on a session key to secure their communications. Nodes i and j can establish a pair-wise key $f(i, j)$ without exchanging any additional packets. To illustrate, node i will determine the shared key by evaluating its polynomial share $f(i, y)$ at point j. In the same manner, node j will perform a similar operation to reveal the same secret key. This approach outperforms Blom's method in two ways: it eliminates the necessity to broadcast public information in the neighbour discovering step and avoids the need to execute vector multiplications in the key establishment step. These two points lead to a significant reduction in power requirements in the sensor nodes. Regarding the use of memory space, each node in this scheme must store a t-degree polynomial which occupies $(t + 1)\, log(q)$ storage units.

Like Blom's scheme, capturing fewer than t sensor nodes does not have any impact on the link security of non-compromised nodes [36]. On the other hand, choosing a high value of t is not recommended because this will consume considerable amount of node memory. Therefore, as the number of sensor nodes in a network increases, adversaries have a greater opportunity in compromising more than t nodes, thereby successfully breaking the security of the entire network. To enhance both scalability and resistance, Liu and Ning in [37] combined the polynomial-based key pre-distribution scheme in [36] with the *basic scheme* in [27]. Rather than exploiting a single symmetric polynomial, several bivariate t-degree polynomials are utilized to construct the pool of secret keys. Prior to distributing sensor nodes, trusted base stations generate an arbitrary set F of symmetric bivariate t-degree polynomials over $GF(q)$. For each sensor node, trusted base stations randomly select a subset F_i of polynomials obtained from the general

set F ($\mathrm{F_i} \subset \mathrm{F}$), and preload this subset into node memory. For local neighbours in a surrounding area, each node will identify those with which it shares at least a single polynomial. Then both nodes will follow the same procedure as explained in [36]. If a pair of adjacent nodes does not have a common polynomial, they will seek the help of intermediate nodes which share at least a single polynomial with both nodes. Compared with the method in [36], the scheme proposed by Liu and Ning provides strong resilience against node capture, especially in cases when the sensor application ensures that a particular polynomial is not used more than t times.

2.6 Public Key Schemes

The pre-distribution approaches discussed earlier provide partial solutions to the key agreement problem with respect to robustness of cryptographic keys, efficient use of physical resources, and scalability of the underlying networks. For this reason, several attempts have been made to make public key schemes feasible for use in wireless sensor networks. For instance, Gaubatz, Kaps, and Sunar in [38] demonstrate the possibility of implementing a public-key encryption technique whose power consumption is less than 20 microwatts. In order to achieve this goal, the authors examined two algorithms: Rabin and NTRU-Encrypt. The former algorithm is a customized version of the well known RSA algorithm where the major modification is fixing the exponents to the value 2. Studying Rabin's scheme [38], it was found that the cost of decrypting a packet is expensive for sensor nodes as this operation consumes significant power compared with that used normally in encryption operations. The reason is that Rabin's scheme is based on the factorization of large numbers. Hence, this algorithm is predominantly suitable for asymmetric sensor networks in which nodes perform encryption and create signatures and base stations are responsible for decryption. Otherwise, Rabin's scheme is an infeasible solution for

wireless sensor networks. Because of this, the authors moved their focus to the NTRU-Encrypt algorithm, which is based on the hardness of the closest vector problem in high dimensional lattices. The authors in [38] implemented an NTRU-Encrypt algorithm using 3000 gates in order to provide a key agreement scheme whose power consumption is lower than 20 microwatts.

Recently, Elliptic Curves Cryptography (ECC) has drawn the attention of many researchers in the field of sensor networks due to the fact that it outperforms RSA from three perspectives: fewer computational instructions, less memory usage, and a key size of 160 bits providing the same security level as RSA with a key of 1024 bits. Indeed, work has been done to optimize elliptic curve algorithms such that they become suitable for wireless sensor networks. Preliminary work was done in [39] where a hybrid of symmetric and elliptic curve cryptography was utilized. However, it was assumed in [39] that the sensor nodes in the network have physical hardware consisting of a 16-bit microcontroller running at a clock rate of 16 megahertz. As a consequence, the results in [39] could be misleading because sensor nodes typically do not possess such sophisticated resources as discussed in Chapter 1. For sensor nodes with a 7.3828MHz 8-bit microcontroller, an implementation of elliptic curve cryptography was presented by Malan et al. in [40]. Here, a pair of sensor nodes was able to share a secret key in approximately 65 seconds using a practical amount of memory, specifically around one kilobyte of static RAM in addition to approximately 34 Kilobytes of ROM. To provide a better implementation, Blaß and Zitterbart in [44] suggest pre-computing a number of frequent operations and storing these parameters on external flash memory. For example, as point multiplication in an elliptic curve algorithm uses a large fixed matrix, this matrix can be constructed in advance and preloaded into the additional memory space of every sensor node. This will conserve the power supplies of sensor nodes and result in increased network lifetime.

Besides optimizing the program code for elliptic curve cryptography, it would be useful to embed a low-power processor that is specialized in performing the core operations used frequently by elliptic curve algorithms. This will help in the effective execution of complex operations, such as scalar-point multiplication, and enable nodes to exploit stronger cryptography without a significant impact on energy resources. For example, the authors in [41] constructed a low-power processor whose architecture was fabricated using 18720 gates in 0.13 μm CMOS technology. This processor implemented an elliptic curve algorithm over the finite field $GF(2^{100}+1)$ and used less than 400 microwatts of power at a 500 kilohertz operating frequency. To further reduction in the power consumption, a low-power processor supported by a highly optimized modular Arithmetic Logic Unit (MALU) was used in [42] to implement elliptic curve cryptography over the finite field $GF(2^{131})$. As compared with [41], the method in [42] succeeded using 6718 gates not only in decreasing the consumed power to less than 30 microwatts but also in improving the cryptographic strength.

2.7 Non-interactive Identity-based Schemes

An important improvement in the implementation of elliptic curve cryptography is a design in which sensor nodes exchange their public information without any communication. Jing et al. in [45] applied this concept on large-scale wireless sensor networks by proposing an identity-based elliptic curve algorithm. In their method, a sensor node exploits the addresses of its neighbours in order to obtain their public keys. This facilitates the process of generating cryptographic keys for communication sessions. Eliminating the necessity of certificates in a network enhances the performance of the key agreement step because no energy is required to forward and verify public keys. To illustrate, the identity-based approach in [45] reduces the

power consumption for packet transmission by 28.5% and reduces the energy needed in similar schemes such as simplified SSL using an abbreviated certificate by almost 35%.

Another non-interactive identity-based scheme can be found in [46] where the concept of pairing based cryptography is utilized. Pairing based algorithms can be explained as follows. Prior to the deployment phase, a trusted base station randomly chooses a master key K. For each sensor node in the network, base stations map the node identity to a point on an elliptic curve using a hashing-and-mapping function Q in order to determine the node public key

$$Public_i = Q(ID_i) \quad (6)$$

To compute the private key of a sensor node, base stations multiply the node public key by the general master key

$$Private_i = [K] \times Public_i \quad (7)$$

To finish the initialization stage, each sensor node i is equipped with its security credentials including: unique identification, exclusive private key, and the general hashing-and-mapping function. Once the sensor nodes are scattered in the field of interest, every node discovers the neighbours within its transmission coverage. Two neighbouring nodes can establish a secure radio connection using the pairing function whose main properties are: 1) it is symmetric so that $\big(F(P,Q) = F(Q,P)\big)$ is satisfied, and 2) it is bi-linear so that $(F([a]P,[b]Q) = F(P,[b]Q)^a = F([a]P,Q)^b = F(P,Q)^{ab})$ is satisfied. To clarify the key establishment step, sensor node A extracts the public key of its neighbour B by inputting the identity of the latter in the mapping function Q as shown in (6), and vice-versa. Without performing any data interaction, both sensor nodes A and B can derive a shared session key as illustrated in Figure 4. According to Oliveira et al. [46], pairing based cryptography allows sensor nodes to provide an acceptable level of

security and perform key agreement within about five and half seconds using a 7.3828MHz 8-bit ATmega128L microcontroller [46].

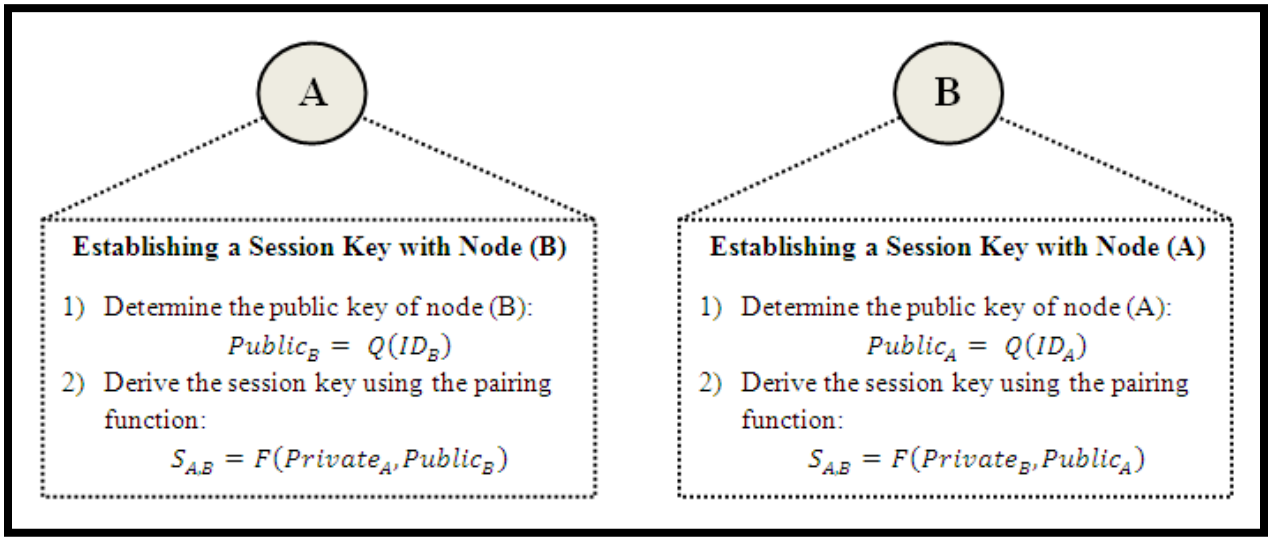

Figure 4: A pair of sensor nodes establishing a session key.

2.8 Discussion

As mentioned in Chapter 1, this thesis contributes to the field of key agreement by introducing three schemes whose memory, processing, and communication requirements are low. These methods utilize the concept of third parties to reduce the cryptographic burden of public-key based schemes and the key management overhead of symmetric key approaches. To achieve this, the proposed methods replace high cost public-key operations at the sensor nodes with a few simple hash operations, that is, nodes use one-way functions in which the input is data of arbitrary length and the output is a unique value of a specific size [24]. Furthermore, additional nodes called third parties are deployed in the network. These assisting nodes do not perform sensing, routing or packet forwarding; they are only responsible for pair-wise key establishment between sensor nodes. All three approaches have many advantages over existing techniques. For instance, a sensor node using these schemes needs to make just a few local

contacts to establish a pair-wise connection with its neighbours with very high probability. Moreover, the majority of sensor nodes must store only a small number of keys in their memory. Besides these benefits, the methods introduced here include an authentication mechanism to prevent impersonation attacks.

To the best of our knowledge, the idea of exploiting third parties in wireless sensor networks has been discussed only in [47] where Dong and Liu used auxiliary sensors for pair-wise key establishment. Algorithm 2.1 explains their proposed scheme in detail. Clearly, the method developed in [47] is resistant to node capture because private secrets are only known by their corresponding nodes, whereas assisting nodes store only public information, including the hash images $H(K_i, id_i)$. Furthermore, sensor nodes often employ more than one assisting node to help in establishing session keys with their neighbours. Hence, the session keys generated may not be impacted by the compromise of a few assisting nodes. However, the approach in [47] suffers from two important disadvantages: it is not scalable to redistribute additional nodes after the deployment stage, and it requires a massive amount of memory in the assisting nodes. Even though Dong and Liu state that assisting nodes can utilize all their memory to store the hash images of sensor nodes, this assumption may not be sufficient for their method to be feasible for sensor networks. To illustrate, *Telos mote* needs its entire 1MB of flash memory to store the hash images for only 65536 sensor nodes, assuming the hash function provides a 16 Bytes long value. An upper limit for the number of nodes makes the network non-scalable and so this scheme is impractical for large networks. To overcome this limitation, we propose in the next chapters three scalable and efficient key agreement methods in which both third parties and sensor nodes use a small number of memory units in the key establishment step.

Algorithm 2.1: pair-wise key establishment using auxiliary sensors node

- **Before the deployment stage:**
  1. A trusted base station generates a random key K for each sensor node in the network and preloads this key as private information into the node's memory.
  2. It also performs the hash function $H(K_i, ID_i)$ for each node and stores the resulting values associated with the node identification in the assisting nodes.
- **The key agreement stage:**
  1. Node A discovers its neighbours in addition to all assisting nodes in its transmission range.
  2. Then node A sends a request to the discovered assisting nodes in its neighbourhood stating its desire to establish a secure radio connection with node B, as an example. To authenticate this step, the sensor node A protects its outgoing message with the key $H(K_A, ID_A)$.
  3. Responding to such request, an assisting node generates a random key and then constructs two packets containing this random key: one is protected by $H(K_A, id_A)$ and the other by $H(K_B, id_B)$, similar to what is performed in *Needham-Schroeder symmetric key protocol*.
  4. After receiving the expected responses, node A combines all messages received by the assisting nodes and sends the obtained key to its neighbour B:

     $$K_{A,B} = R_1 \oplus \ldots \oplus R_r$$

     Exploiting more than one assisting node can increase the security of the wireless sensor network since the resulting key can be secure even though some assisting nodes are compromised.
  5. Node A repeats both steps 2 and 4 for each neighbour.
  6. It is possible that a sensor node has no assisting node in its neighbourhood. In this scenario, the neighbours of node A can help in finding an assisting node within a pre-defined number of hops and then forward the responses back to node A.

Chapter 3

Key Establishment Using Third Parties

3.1 Introduction

In this chapter, we propose a key agreement scheme in which sensor nodes establish secure radio connections with their neighbours relying on the assistance of third parties. These third parties are only responsible for the pair-wise key establishment among sensor nodes, so they do not perform any other operations such as sensing or packet routing. We will start our discussion in this chapter by describing the proposed method with respect to three points: how secret shares are distributed, how local neighbours are discovered, and how secure channels are created. Then we will analyze the method's performance using five metrics: local connectivity, resilience against node capture, memory usage, communication overhead, and computational overhead.

3.2 An Efficient Key Agreement Scheme Using Third Parties

Like random key pre-distribution schemes, the method introduced here consists of three main steps: distributing secret shares, discovering local neighbours, and establishing secure channels. Assume t third party nodes in addition to n sensor nodes are deployed uniformly in the field of interest. Prior to deployment of the sensor network, a trusted base station generates a random encryption key S as well as a random authentication key A. Then the trusted base station will store both keys as private information into the memory units of all t third parties. Furthermore, every sensor node in the network is equipped with unique encryption and

authentication keys. To do so, the trusted base station computes for each node i two values

$$\begin{cases} S_i = Hash(S, ID_i) \\ A_i = Hash(A, ID_i) \end{cases} \quad (8)$$

and preloads these two keys into the node memory unit.

After scattering sensor nodes in the target region, the nodes perform a discovery process in order to find their local neighbours as well as their closest third party. Finding local neighbours is a simple operation since sensor nodes periodically broadcast HELLO packets or what is called beaconing to advertise their existence in a neighbourhood. Therefore, nodes can easily depend on these packets to identify adjacent nodes. To protect the network from HELLO flood attacks as described in Chapter 1, a pair of neighbouring nodes performs a two-way handshake to ensure that the other node is located within its radio range. On the other hand, it is not sufficient to allow third parties to advertise their existence in a neighbourhood relying on HELLO packets alone. The reason is that malicious nodes can spoof the contents of a HELLO packet originating from a genuine third party and then forward this packet to the sensor nodes in its transmission range. Innocently, these sensor nodes can be deceived to assign this malicious node as their closest third party node, thereby disrupting the functionality of the proposed method.

To verify the legitimacy of a third party in a sensor network, we propose two novel authentication mechanisms: white list and hash chain schemes. In the white list method, a trusted base station creates a predefined list of identities belonging to authorized third parties which a sensor node can use to establish secure connections with its neighbours. This predefined list is preloaded into the memory unit of every node in the network. Upon receiving a HELLO packet from a third party, sensor nodes look up its identity in the predefined list in order to determine whether it is a legitimate third party or not. Even though the white list method provides a simple

solution, it may not be practical with regard to scalability and memory usage for some applications. For example, sensor nodes in networks where the number of third parties is high consume a considerable amount of memory resources to store the predefined list of identities so that the white list scheme can be implemented. Moreover, it is anticipated that the node failure rate in sensor networks is high since they often operate in outdoor environments. Thus, it may be necessary to redeploy additional third parties in order to replace the failed ones and improve network performance. This means that after the redeployment process sensor nodes may encounter a legitimate third party that is not recorded in the white list.

To overcome these drawbacks, we will utilize a mechanism called the hash chain method. This provides a simple and scalable solution without consuming substantial storage resources in the sensor nodes. In this method, sensor nodes need to store just a single key in their memory units. Also, they can easily accommodate new legitimate third parties which are redistributed in the field of interest after the deployment phase. To implement the hash chain mechanism, a trusted base station randomly generates a key M and then inputs this generated key in a sequence of hash functions as shown in Figure 5. Prior to deployment, third parties are given the chain of keys $L = \{L_0, L_1, L_2, \dots, L_{a-1}, L_a\}$, while sensor nodes are equipped with a single key L_a. Once the sensor nodes are spread in the target region, third parties advertise their existence in a neighbourhood through broadcasting a customized HELLO packet containing the key L_{a-1}. Upon receiving a HELLO packet, every sensor node in the network checks the legitimacy of the third party as follows. Sensor nodes extract the key L_{a-1} from the incoming HELLO packet and then input L_{a-1} to the predefined hash function. If the value resulting from the hash function $H(L_{a-1})$ matches the key L_a stored in node memory, authentication is successful. In this case, each sensor node should update its stored key by replacing L_a with L_{a-1}. Next time, the third

party will unitize the key L_{a-2} in its customized HELLO packets, and so on. It is important to note that the value of the parameter a should be chosen carefully taking into account the strength of the security protocols needed in addition to the expected lifetime of the sensor network. For instance, the parameter a used in long-term battlefield surveillance should be greater than the value utilized in applications designed to obtain statistics about the number of visitors in a short-term tourist attraction.

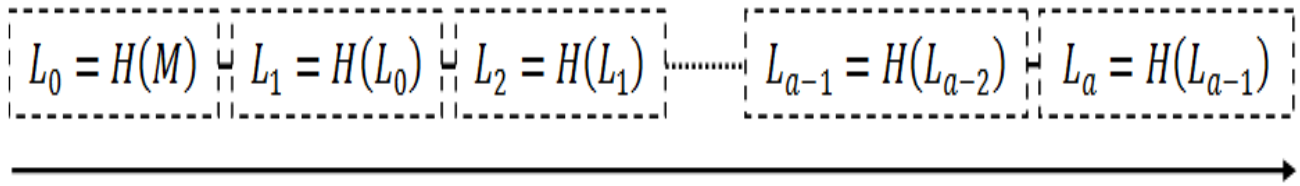

Figure 5: An example of a hash chain generated from a random key *M.*

Once the discovery stage is completed, every sensor node i in the network sends a request to the discovered third party in its transmission range stating its desire to establish secure channels with neighbouring nodes $\{j_1, j_2, \dots, j_d\}$. To help the third party in verifying the identity of the packet originator, node i encrypts its outgoing message with the authentication secret A_i

$$\text{Node(i)} \rightarrow \text{Third Party:} \qquad \mathrm{E}_{\mathrm{A_i}}\big(\mathrm{Request(i, \{j_1, j_2, \dots, j_d\})}\big)$$

Responding to such a request, the third party starts the process of key generation by computing the values S_i and A_i using (8). For each neighbour mentioned in the received request, the third party determines an encryption key S_{j_x} and then calculates a temporary secret share for use in generating a session key between both the sensor node i and its neighbour j_x as follows

$$Secret(i, j_x) = Hash\big(S_i, ID_{j_x}\big) \oplus Hash\big(S_{j_x}, ID_i\big) \qquad (9)$$

The third party sends back to the sensor node i its temporary secret shares protected with the authentication key A_i as shown below

$$Third\ Party \rightarrow Node(i): \qquad E_{A_i}\big(i \,||\, j_x \,||\, Secret(i, j_x)\big) \qquad for\ x = 1, \dots, d$$

Upon receiving these temporary secret shares, the sensor node i uses the identity of its neighbouring node j_x in order to determine a session key between the two nodes given by

$$Session_{j_x} = Secret(i, j_x) \oplus Hash(S_i, ID_{j_x}) \quad (10)$$

Although sensor nodes i and j_x can rely on this session key to secure their radio connections, we recommend that sensor node i generates a new random secret key K_{i,j_x} and then constructs a packet containing this random key. This packet should be encrypted with the session key obtained in (10) and then forwarded to the neighbouring node j_x. Performing this additional step boosts the network security and prevents any negative impact from the disclosure of session keys after a third party is compromised. As soon as the neighbouring node j_x receives the encrypted packet, it inputs the identity of node i into the predefined hash function in order to extract the session key used to encrypt the received packet as follows

$$Session_i = Hash(S_{j_x}, ID_i) \quad (11)$$

Obviously, the session keys in (10) and (11) are identical. To prove this point, we can easily substitute (9) into the right hand of (10) yielding (11) as shown below

$$Session_{j_x} = Secret(i, j_x) \oplus Hash(S_i, ID_{j_x})$$

$$Session_{j_x} = \left(Hash(S_i, ID_{j_x}) \oplus Hash(S_{j_x}, ID_i)\right) \oplus Hash(S_i, ID_{j_x})$$

$$Session_{j_x} = \left(Hash(S_i, ID_{j_x}) \oplus Hash(S_i, ID_{j_x})\right) \oplus Hash(S_{j_x}, ID_i) = Hash(S_{j_x}, ID_i)$$

$$Session_{j_x} = Hash(S_{j_x}, ID_i) = Session_i$$

At this point, the neighbouring node j_x uses the session key in (11) to decrypt the received packet and then obtain the secret key K_{i,j_x}. Sensor nodes i and j_x can then employ the secret key K_{i,j_x} to secure their radio channel.

Since it is possible to encounter situations in which a sensor node i has no third parties in its local neighbourhood, it is necessary to include an additional step in the key establishment phase to avoid such scenarios. In this step, a neighbour of node i will help him in finding a third party within a predefined number of hops and then communicate with this third party on behalf of node i. Similar to what was explained earlier, sensor node i will send a request to a neighbour indicating its desire to establish secure channels with nodes $\{j_1, j_2, \dots, j_d\}$. This request will be encrypted with the authentication secret A_i in order to protect the contents of the packet and confirm the identity of the originator. The neighbouring node will forward this request to a nearby third party in its local transmission range. Upon receiving the anticipated responses from the third party, the neighbouring node will forward these responses to node i. Once the temporary secret shares are received, node i will continue the process of key establishment so that secure channels are created with its adjacent nodes.

3.3 Local Connectivity

The local connectivity of a sensor network is typically represented by the probability of two neighbouring nodes being able to find a common secret key. Since sensor nodes in our proposed method depend on the assistance of a third party to generate shared keys with their neighbours, the definition of local connectivity should be different. In other words, a pair of neighbouring nodes can establish a secure radio connection if one of the two has a third party in its transmission range. Consequently, the local connectivity of a sensor network in the proposed method is the probability that one of two neighbouring nodes discovers a third party in its neighbourhood. Throughout this thesis, the term p_{Local} refers to the local connectivity of a sensor network. To compute p_{Local}, it is necessary to: a) determine the expected area of coverage for

two adjacent nodes in the deployment region, and b) calculate the probability that at least one third party is located within this area. In the following sections, we assume that sensor nodes in addition to third parties are distributed uniformly over a field of interest whose size is equal to G.

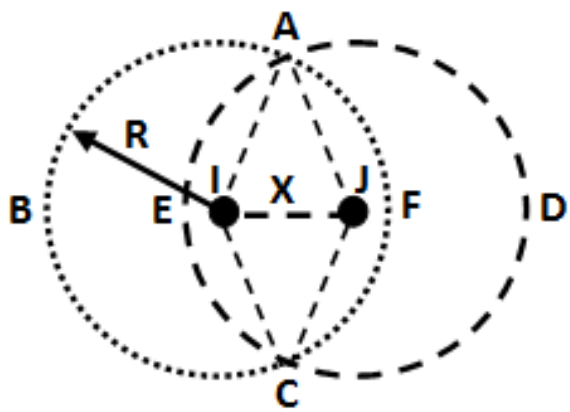

Figure 6: A pair of neighbouring nodes in a wireless sensor network.

a) The Expected Area of Coverage for Two Adjacent Nodes in the Deployment Region

Figure 6 shows a pair of neighbouring nodes in a wireless sensor network. According to Chan et al. [31], the expected area of coverage for the neighbouring nodes i and j is

$$\check{E}(x) = \int_0^R Area_{ABCD}(x)\, f(x)\, dx \qquad (12)$$

where $Area_{ABCD}(x)$ represents the area of both circles minus the overlapped region $AECF$, R indicates the transmission radius of a sensor node, and $f(x)$ is the probability density function of the sensor node distribution in the field of interest. Because sensor nodes and third parties are distributed uniformly in the deployment area, $f(x)$ is calculated as follows [48]

$$f(x) = \frac{dF(x)}{dx} = \frac{d}{dx}\big(P(distance < x)\big) = \frac{d}{dx}\left(\frac{\pi x^2}{\pi R^2}\right) = \frac{2x}{R^2} \qquad (13)$$

Regarding $Area_{ABCD}(x)$, we must first compute the overlapped region $AECF$ which is given by

$$Area_{AECF}(x) = Sector_{AIC} + Sector_{AJC} - Triangle_{AIJ} - Triangle_{CIJ} \qquad (14)$$

$$Area_{AECF}(x) = 2\,R^2\,cos^{-1}\left(\frac{x}{2R}\right) - x\sqrt{R^2 - \frac{x^2}{4}} \quad (15)$$

Subtracting the overlapped region from the area of both circles in Figure 6 yields

$$Area_{ABCD}(x) = 2\pi R^2 - 2\,R^2\,cos^{-1}\left(\frac{x}{2R}\right) + x\sqrt{R^2 - \frac{x^2}{4}} \quad (16)$$

On this ground, the expected area of coverage for sensor nodes i and j is given by

$$\breve{E}(x) = \int_0^R Area_{ABCD}(x)\,f(x)\,dx$$

$$= \int_0^R \left(2\pi R^2 - 2\,R^2\,cos^{-1}\left(\frac{x}{2R}\right) + x\sqrt{R^2 - \frac{x^2}{4}}\right)\left(\frac{2x}{R^2}\right)dx \quad (17)$$

$$\breve{E}(x) = (2\pi x^2)|_0^R + \left(Rx\sqrt{4 - \frac{x^2}{R^2}}\right)\Bigg|_0^R + \left(2R^2\,tan^{-1}\left(\frac{x}{\sqrt{4R^2 - x^2}}\right)\right)\Bigg|_0^R$$

$$+ \left(\frac{\sqrt{4R^2 - x^2}}{4R^2}(x^3 - 2R^2x)\right)\Bigg|_0^R - \left(4R^2\,sin^{-1}\left(\frac{x}{2R}\right)\right)\Bigg|_0^R - \left(2x^2\,cos^{-1}\left(\frac{x}{2R}\right)\right)\Bigg|_0^R$$

$$\breve{E}(x) = 1.413497\pi\,R^2 \quad (18)$$

After determining the expected area of coverage for a pair of adjacent nodes, we next calculate the probability that at least one third party is located within the region $Area_{ABCD}(x)$.

b) The Probability of Finding at Least One Third Party in the Region $Area_{ABCD}(x)$

Since third parties are uniformly deployed in the wireless sensor network, the probability that a third party node is located inside the region $Area_{ABCD}(x)$ is given by

$$p = \frac{\text{The expected region } Area_{ABCD}(x)}{\text{The size of the deployment area}} = \frac{\breve{E}(x)}{G} = \frac{1.413497\pi\,R^2}{G} \quad (19)$$

Because it is possible that more than one third party is located in a particular area, the binomial distribution can be used to derive the probability that z third parties are within the region $Area_{ABCD}(x)$. This probability is given by

$$p(z = \check{z}) = \binom{t}{\check{z}} p^{\check{z}}(1-p)^{t-\check{z}} \quad (20)$$

As mentioned earlier, a pair of neighbouring nodes can establish a secure radio connection when one of the two nodes has at least one third party in its transmission range. Based on this, we can conclude that the local connectivity of a sensor network in the proposed method is given by

$$p_{Local} = p(z \geq 1) = 1 - p(0) = 1 - \left(\binom{t}{0} p^0 (1-p)^t\right) = 1 - (1-p)^t \quad (21)$$

To simplify (21), let d be the average number of neighbours within the radio coverage of a sensor node. When $n \gg d$ and $G \gg \pi R^2$, we can assume

$$\frac{\pi R^2}{G} = \frac{d}{n} \quad (22)$$

Substituting (19) and (22) into (21), the local connectivity can be rewritten as follows

$$p_{Local} = 1 - (1-p)^t = 1 - \left(1 - \frac{1.413497\pi\, R^2}{G}\right)^t = 1 - \left(1 - \frac{1.413497\, d}{n}\right)^t \quad (23)$$

Figure 7 shows the local connectivity of a network when the number of sensor nodes is equal to 10000 with various densities, i.e., the average number of neighbours in a node's transmission range. Because the number of third party nodes plays an important role in determining the local connectivity of a network, we plot the local connectivity with respect to the ratio of the number of third parties to the number of sensor nodes. Clearly, increasing this ratio to more than 40 percent will result in perfect connectivity, but this is not feasible from both a practical and economical points of view. As a consequence, we consider lower ratios such as 10 percent which provides reasonable connectivity. As shown in Figure 7, the proposed method

with a 10 percent ratio gives approximately 92.5 percent of local connectivity even in low dense networks like when $d = 20$.

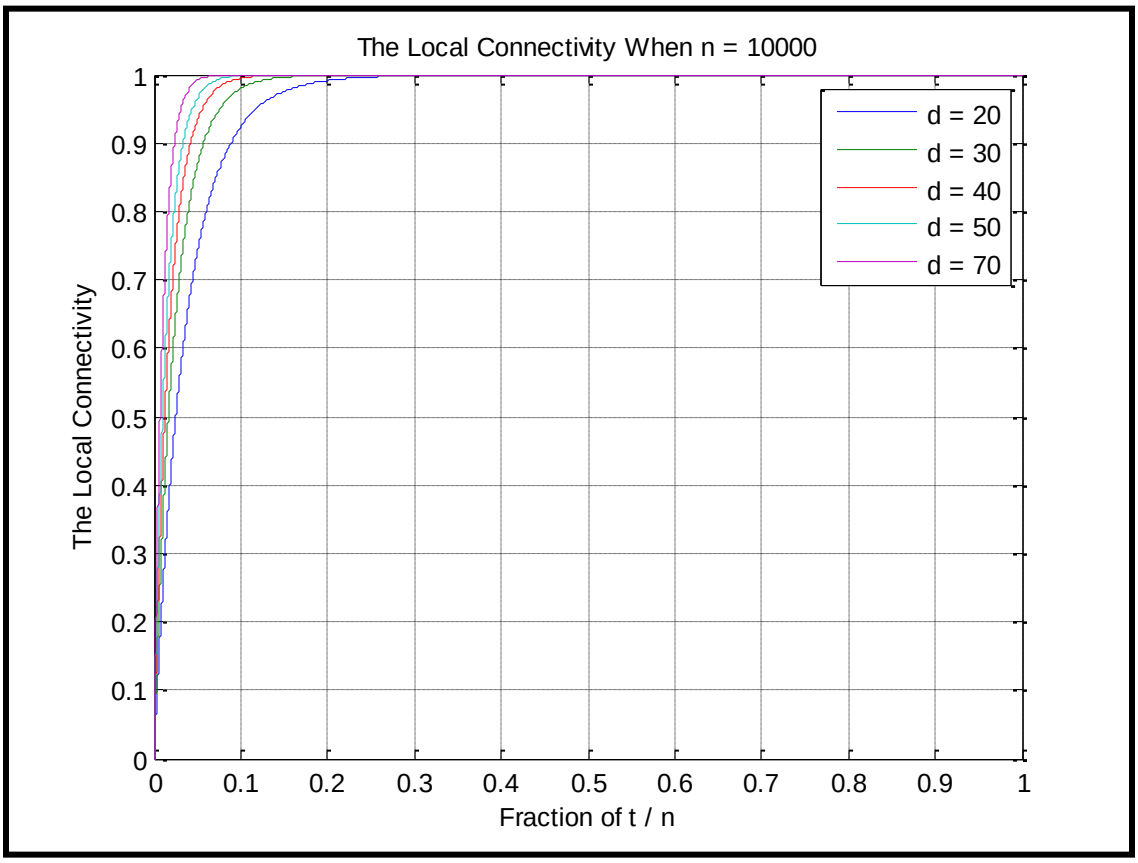

Figure 7: The local connectivity of a sensor network applying the proposed method.

Figure 7 also shows it is possible that a pair of neighbouring nodes may have no third party in their transmission range. As a result, it will be necessary to depend on an intermediate node in order to complete the key establishment process, as it was described previously. Since the communication overhead is an important concern in wireless sensor networks, we should limit the number of intermediate nodes between a sensor node and a third party to only one. In other words, if two neighbours are trying to establish a pair-wise key to secure their radio channel, one of the two nodes should discover a third party within two hops as shown in Figure 8. Note that a two hop distance does not necessary mean a radius of $2R$. This value is quite

optimistic in non-dense sensor networks as it would require the intermediate node to be located on the edge of a neighbouring node boundary. To compute the local connectivity through an intermediate node in a non-dense sensor network, a good approximation for the two hop range is a radius of $3R/2$ rather than $2R$.

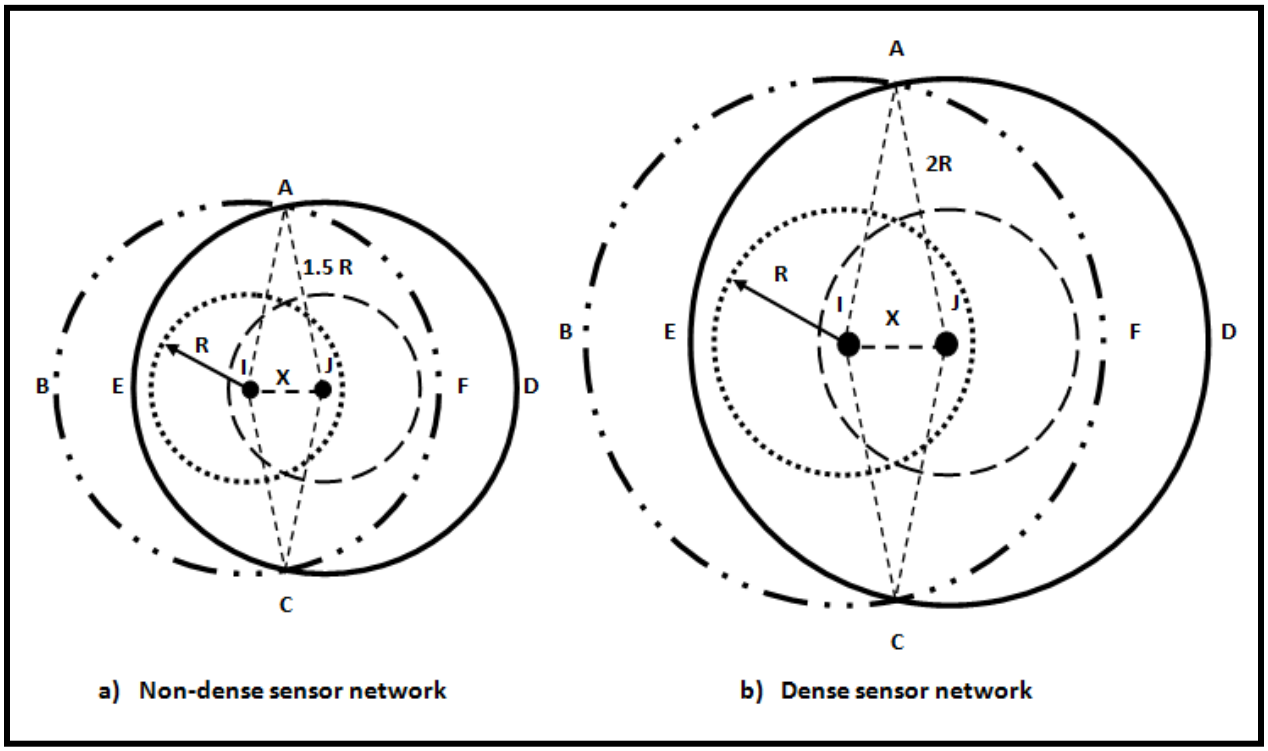

Figure 8: The scope of discovery possible for two neighbouring nodes.

Allowing sensor nodes to seek assistance from an intermediate node to discover a third party can increase the local connectivity of a sensor network. To calculate p_{Local} in this case, we simply need to compute the expected area of $Area_{ABCD}$ in Figure 8 and then determine the probability that at least one third party is located inside this region. In situations where sensor nodes are deployed in a non-dense network, the region $Area_{ABCD}$ represents the area of both circles in Figure 8.a minus their overlapped region $AECF$. In fact, $Area_{ABCD}$ is given by

$$Area_{ABCD}(x) = 4.5\pi R^2 - 4.5\, R^2\, cos^{-1}\left(\frac{x}{3R}\right) + x\sqrt{\frac{9}{4}R^2 - \frac{x^2}{4}} \qquad (24)$$

Therefore, the expected area of coverage for two neighbouring nodes in a non-dense network is

$$\check{E}(x) = \int_0^R Area_{ABCD}(x)\, f(x)\, dx$$

$$= \int_0^R \left(4.5\pi R^2 - 4.5\, R^2\, cos^{-1}\left(\frac{x}{3R}\right) + x\sqrt{\frac{9}{4}R^2 - \frac{x^2}{4}}\right)\left(\frac{2x}{R^2}\right) dx = 2.87947\pi\, R^2$$

Substituting the expected area obtained in the previous equation into (19), the probability that a third party node is located inside the region $Area_{ABCD}(x)$ becomes as follows

$$p = \frac{\text{The expected region } Area_{ABCD}(x)}{\text{The size of the deployment area}} = \frac{\check{E}(x)}{G} = \frac{2.87947\pi\, R^2}{G}$$

Similar to what was done in (20), the binomial distribution can be used in order to determine the probability that at least one third party is located within the region $Area_{ABCD}(x)$. Based on this, the local connectivity of the non-dense sensor network in Figure 8.a is given by

$$p_{Local} = 1 - (1-p)^t = 1 - \left(1 - \frac{2.87947\check{n}}{n}\right)^t \qquad (25)$$

On the other hand, the region $Area_{ABCD}$ in the dense sensor network in Figure 8.b is as follows

$$Area_{ABCD}(x) = 8\pi R^2 - 8\, R^2\, cos^{-1}\left(\frac{x}{4R}\right) + x\sqrt{4R^2 - \frac{x^2}{4}}$$

Consequently, the expected area of coverage for sensor nodes i and j in Figure 8.b is

$$\check{E}(x) = \int_0^R Area_{ABCD}(x)\, f(x)\, dx = \int_0^R \left(8\pi R^2 - 8\, R^2\, cos^{-1}\left(\frac{x}{4R}\right) + x\sqrt{4R^2 - \frac{x^2}{4}}\right)\left(\frac{2x}{R^2}\right) dx$$

$$\check{E}(x) = 4.84349\pi\, R^2 \qquad (26)$$

Substituting (26) into (21) yields the local connectivity of the dense network in Figure 8.b

$$p_{Local} = 1 - (1-p)^t = 1 - \left(1 - \frac{4.84349\pi\, R^2}{S}\right)^t = 1 - \left(1 - \frac{4.84349\check{n}}{n}\right)^t \qquad (27)$$

Figures 9 and 10 show the local connectivity obtained using (25) and (27), respectively. Clearly, the sensor networks in both scenarios achieve high local connectivity when the number of third parties is greater than 9 percent of the total number of sensor nodes in the field of interest.

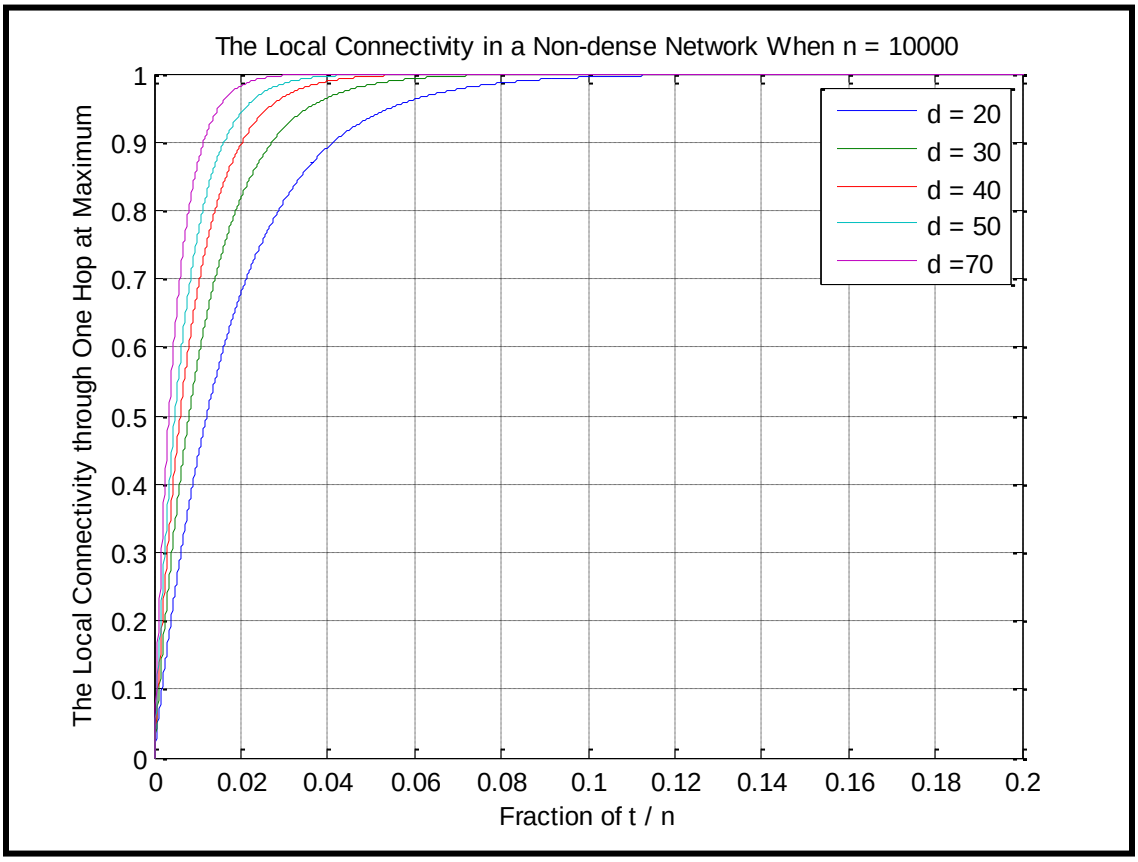

Figure 9: Local connectivity using maximum of one intermediate node in non-dense networks.

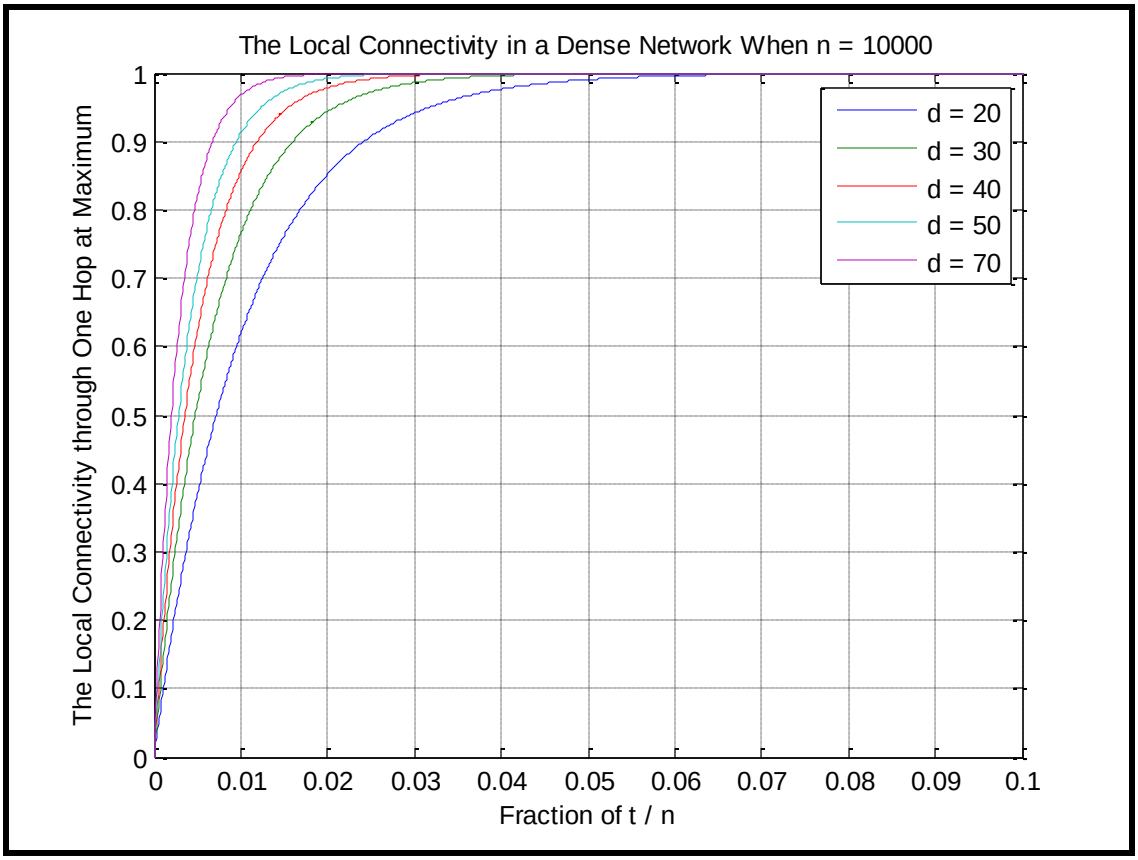

Figure 10: Local connectivity using a maximum of one intermediate node in dense networks.

Previously, three different scenarios were considered in the key establishment phase: a) sensor nodes are not allowed to use any intermediate node, b) sensor nodes are deployed in non-dense networks in which they can use a maximum of one intermediate node, and c) sensor nodes are distributed in dense networks where they can utilize at most one intermediate node. When the average number of neighbours within the transmission range of a sensor node is equal to $d = 20$, Figure 11 as well as Table 3.1 illustrate the improvement obtained using a maximum of one intermediate node in our propose method. For example, the network in scenario (a) needs 4.72 times the number of third parties required in scenario (c) such that a fully connected network is achieved. Compared with the non-dense networks in scenario (b), the network in scenario (a) still requires 2.59 times the number of third parties needed in scenario (b) in order to have a complete

local connectivity. Changing d from 20 to 40, the performance will be enhanced since increasing the expected number of neighbours within a transmission range boosts the probability that a pair of neighbouring nodes can discover a third party. Indeed, Figure 12 in addition to Table 3.2 show a comparison between the local connectivity of the three scenarios when $d = 40$.

t/n	No intermediate nodes are used in the sensor network	At most one intermediate node is used in the non-dense sensor network	At most one intermediate node is used in the dense sensor network
0.50 %	13.14 %	24.97 %	38.39 %
0.75 %	19.00 %	34.94 %	51.55 %
1.00 %	24.44 %	43.55 %	61.85 %
2.00 %	42.60 %	67.77 %	85.17 %
3.00 %	56.16 %	81.40 %	94.13 %
4.00 %	66.34 %	89.15 %	97.63 %
5.00 %	74.02 %	93.61 %	99.03 %
6.00 %	79.86 %	96.19 %	99.59 %
7.00 %	84.31 %	97.71 %	99.83 %
8.00 %	87.72 %	98.61 %	99.93 %
9.00 %	90.34 %	99.15 %	99.97 %
10.00 %	92.37 %	99.47 %	99.99 %
11.34 %	94.40 %	99.72 %	100.00 %
12.00 %	95.18 %	99.79 %	100.00 %
14.00 %	96.91 %	99.92 %	100.00 %
18.00 %	98.67 %	99.99 %	100.00 %
20.71 %	99.22 %	100.00 %	100.00 %
25.00 %	99.65 %	100.00 %	100.00 %
45.18 %	99.99 %	100.00 %	100.00 %
53.85 %	100.00 %	100.00 %	100.00 %

Table 3.1: Comparison of the local connectivity for three different scenarios when d = 20.

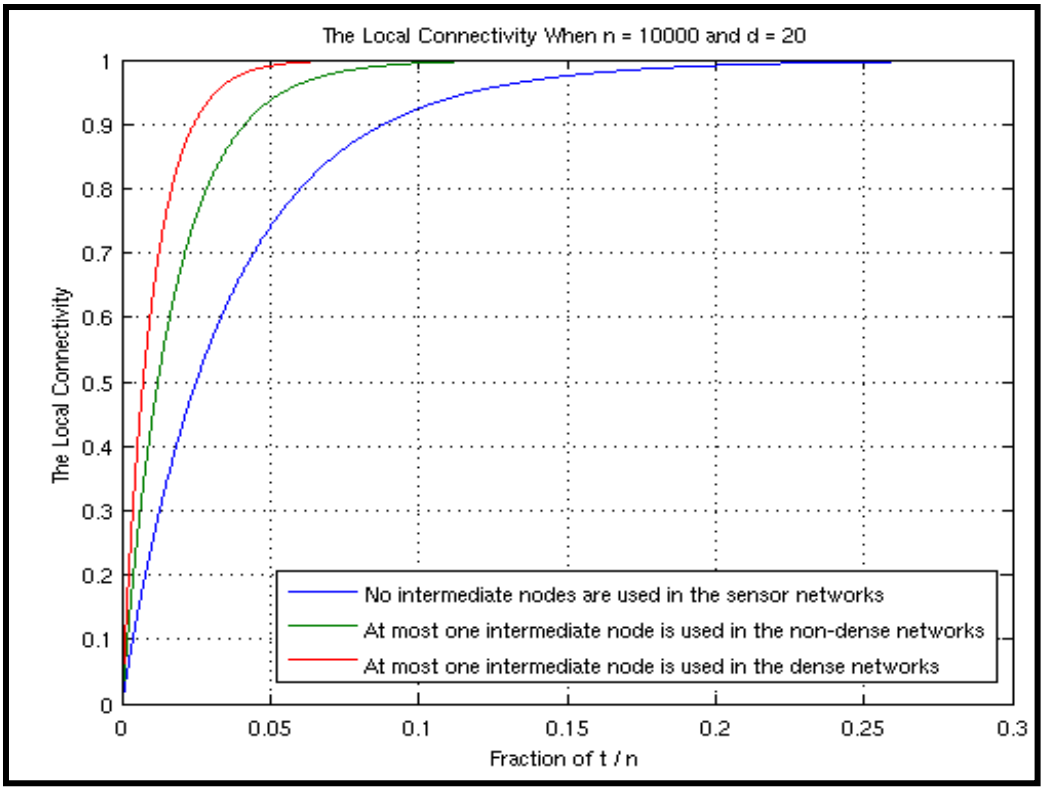

Figure 11: Comparison of the local connectivity for three different scenarios when d = 20.

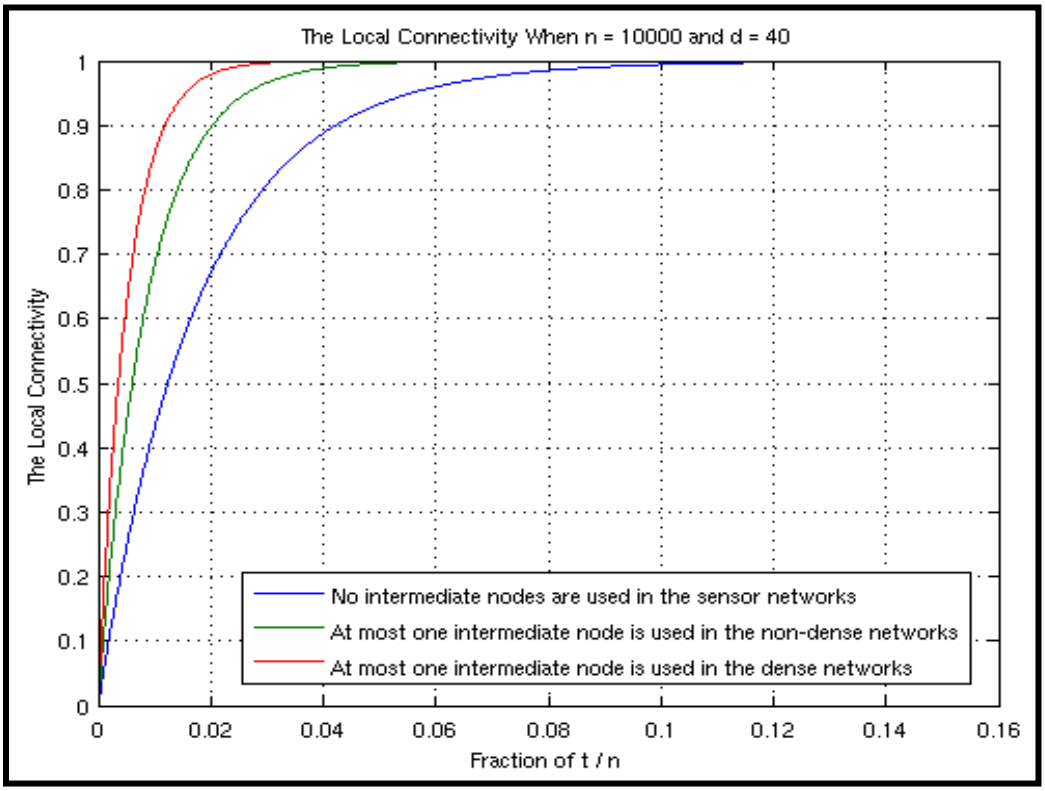

Figure 12: Comparison of the local connectivity for three different scenarios when d = 40.

t/n	No intermediate nodes are used in the sensor network	At most one intermediate node is used in the non-dense sensor network	At most one intermediate node is used in the dense sensor network
0.25 %	13.19 %	25.09 %	38.61 %
0.50 %	24.58 %	43.80 %	62.22 %
0.75 %	34.43 %	57.78 %	76.69 %
1.00 %	42.96 %	68.24 %	85.58 %
2.00 %	67.10 %	89.68 %	97.84 %
3.00 %	80.82 %	96.57 %	99.66 %
4.00 %	88.70 %	98.84 %	99.95 %
5.00 %	93.28 %	99.60 %	99.99 %
5.34 %	94.35 %	99.72 %	100.00 %
6.00 %	95.96 %	99.86 %	100.00 %
7.00 %	97.55 %	99.95 %	100.00 %
8.00 %	98.50 %	99.98 %	100.00 %
9.00 %	99.07 %	99.99 %	100.00 %
9.36 %	99.22 %	100.00 %	100.00 %
10.00 %	99.42 %	100.00 %	100.00 %
11.00 %	99.64 %	100.00 %	100.00 %
12.00 %	99.77 %	100.00 %	100.00 %
13.00 %	99.85 %	100.00 %	100.00 %
14.00 %	99.91 %	100.00 %	100.00 %
18.40 %	99.99 %	100.00 %	100.00 %
21.18 %	100.00 %	100.00 %	100.00 %

Table 3.2: Comparison of the local connectivity for three different scenarios when d = 40.

3.4 Resilience against Node Capture

When an adversary captures a node in a wireless sensor network, he is able to extract the key material, observation data, and software stored on that node. Under this circumstance, it is important to employ key agreement schemes for which a compromised sensor node has a limited impact on the security of the entire network. In other words, the secret keys of non-compromised nodes should not be revealed when a sensor node is captured. Usually, a metric called resilience against node capture is utilized in order to measure the resistance of a key agreement scheme to a node being compromised. This metric is calculated as the ratio of the number of compromised nodes to the percentage of insecure radio connections in the network. High resilience against node capture indicates that a compromised sensor node has a low impact on the secrecy of transmission channels belonging to other nodes.

Unlike the majority of random pre-distribution schemes, compromising a sensor node in our proposed method does not have any effect on the secrecy of transmission channels belonging to other nodes in the network. To clarify, sensor nodes in our scheme store only their private information, which is independent from the private data in other nodes. Consequently, the exclusive use of private keys in the network confines the negative impact of a compromised sensor node to the node itself. On the other hand, compromising a single third party may lower, in rare situations, the resistance of the proposed scheme to a node being compromised. To illustrate, as explained in Section 3.2, every pair of neighbouring nodes depends on their closest third party to establish a secure radio connection. Actually, the third party generates a session key which is used by the pair of sensor nodes to exchange a random secret key. Even though it would be easier for the two neighbouring nodes to rely on the session key generated by their closest third party for communications, the proposed scheme makes both nodes secure their

transmission channel with a random secret key of their own. This measure prevents the negative impact caused by the disclosure of session keys after a third party is compromised. Therefore, the secrecy of non-compromised radio connection is preserved. However, it is still possible in some situations that a third party becomes compromised while pairs of sensor nodes are still exchanging their random secret keys. In this scenario, the radio channels of these nodes may be compromised leading to a slight decrease in the network resilience against node capture.

It is important to note that a single compromised third party will reveal the secret information of other third parties in the field of interest. In this case, the cryptographic and authentication mechanisms introduced in Section 3.2 are inadequate to secure the transmission channels of sensor nodes. To overcome this problem, solutions that help in preventing the possibility of compromising a third party can be used. An intuitive solution would be to equip each third party node with tamper-resistant hardware in order to prevent any disclosure of private keys. Although this option is considered in several secure routing algorithms designed for hierarchical sensor networks, we avoid such solutions since these tamper-resistant components are either simple but not secure, or robust but costly [49].

Another reasonable solution to reduce the negative impact of a compromised third party would be deleting the private information stored in the memory units of third parties as soon as the key agreement process is completed. Normally, the key establishment phase is performed in the beginning of a network lifetime, and it is often accomplished within a period of time T_k. Wireless sensor networks are typically static configurations in which nodes do not change their locations after deployment. On this basis, a sensor node is expected to not seek any further assistance from third parties after the period T_k unless new nodes are redistributed in its neighbourhood. Under these conditions, deleting the private information stored on third parties

after T_k and then putting these third parties in sleep mode will reduce the risk of capturing a third party without affecting the local connectivity of a sensor network. Similar to [50], we can also assume that attackers require time larger than T_k in order to capture a third party. With this assumption, our proposed method provides perfect resilience against node capture. When the network redeploys additional nodes in the field of interest, third parties can be awakened to receive from the trusted base station an encrypted packet containing their private information. Completing the key agreement process, these third parties will delete their secret keys within the period T_k and then go back to sleep mode.

We will assume in this chapter that an adversary needs more than T_k seconds in order to compromise a third party node. Conversely, Chapter 4 will present two novel solutions that can accomplish the same objectives without imposing this assumption.

3.5 Memory Usage

Nodes in wireless sensor networks have limited storage resources. For instance, the *Mica mote* developed by the University of California, Berkeley is equipped with two main memory units: 128 KB of RAM to store programming instruction and 512 KB of flash memory to store observation and other data. For this reason, key agreement schemes should efficiently utilize the storage resources available on sensor nodes. In this context, memory usage is a metric defined as the amount of memory that a node requires in order to establish secure transmission channels with its neighbours.

A significant characteristic of the proposed method is its small memory usage in sensor nodes as well as third parties. To illustrate, every sensor node in the network starts the key agreement process by discovering its closest third party. For this step, sensor nodes need to store

just a single key in their memory space so that the hash chain mechanisms can be performed. After discovering neighbouring third parties, nodes should continue the key establishment phase so that their radio connections are secured. To accomplish this, each sensor node is preloaded with unique encryption and authentication keys as shown in (8). Hence, sensor nodes in the proposed scheme must store only three keys.

Similar to sensor nodes, a third party needs to store three secret keys: an encryption key S to assist a pair of neighbours in generating session keys, an authentication key A to identify the originators of packets sent to or from sensor nodes, and a random value M to help sensor nodes in verifying the legitimacy of a third party in their region. In addition to these three keys, a third party should also be equipped with an exclusive pair-wise key used to initiate secure routes to trusted base stations. As a consequence, third parties in the proposed method require four secret keys in order to conduct the key agreement process. Assuming cryptographic keys are 128 bits in length [51], Table 3.3 summarizes the memory usage of the proposed scheme for both sensor nodes and third parties.

Metric	Sensor Nodes	Third Parties
Memory Usage	Three key units = 384 bits	Four key units = 512 bits

Table 3.3: The memory usage for the proposed scheme.

3.6 Communication and Computational Overhead

Another important metric to evaluate the performance of key agreement schemes for wireless sensor networks is the complexity of communication and processing operations. In this regard, our proposed method has many advantages over existing public key and random pre-distribution techniques. For example, it replaces high cost public-key operations at the sensor nodes with symmetric encryption and a few hash operations. Furthermore, a sensor node needs to

conduct just a few local contacts in order to establish pair-wise keys with its neighbours. In this section, the communication and computational overhead of the proposed scheme is analyzed.

To secure the radio connections of a network, sensor nodes and third parties collaborate with each others to generate an exclusive pair-wise key for each pair of neighbouring nodes. Due to this cooperation, some of the computational operations required in the key establishment process can be performed on sensor nodes, while other operations can be executed on third parties. Figures 13 and 14 illustrate the main operations performed by third parties and sensor nodes, respectively. In both figures, we assume that a sensor node is responsible for generating $d/2$ secret keys on average, where d is the expected number of nodes in a neighbourhood. The reason behinds choosing this value is that a pair of neighbouring nodes can establish a secure radio channel if one of the two nodes obtains assistance from a third party in its transmission range. Thus, a sensor node on average is involved in generating half of its secret keys and receives the remaining half from its neighbours. Figures 13 and 14 do not take into consideration the communication overhead in discovering local neighbours. Since this process is an essential step in all routing protocols, it would be reasonable to assume that sensor nodes know the identities of neighbours prior to the beginning of the key agreement phase.

To secure its radio connections, a sensor node as shown in Figure 14 needs to encrypt and send d packets, perform the hash function $(d+1)$ times, receive and decrypt d packets, and use the random number generator $d/2$ times. On the other hand, to help a node in generating session keys with its neighbours, a third party receives and decrypts $d/2$ packets, performs the hash function $\left(d+\frac{d}{2}+3\right)$ times, encrypts $d/2$ packets, and sends $d/2$ packets back to the sensor node. To indicate the computational and communication overhead for these operations, Table 3.4 shows the amount of energy consumed by each operation when a sensor node is equipped with a

4MHz 8-bit Atmel ATmega128L microcontroller and a 915MHz low-power radio transceiver [52-53].

Operation	Energy Consumed
Encryption using AES-128	1.62 μJ/Byte
Decryption using AES-128	2.49 $\mu J/Byte$
Hashing using SHA-1	5.90 $\mu J/Byte$
Generating a cryptographic key	11.4 $\mu J/Byte$
Receiving a packet	28.6 $\mu J/Byte$
Transmitting a packet	59.2 $\mu J/Byte$

Table 3.4: The energy consumed by computational and communication operations.

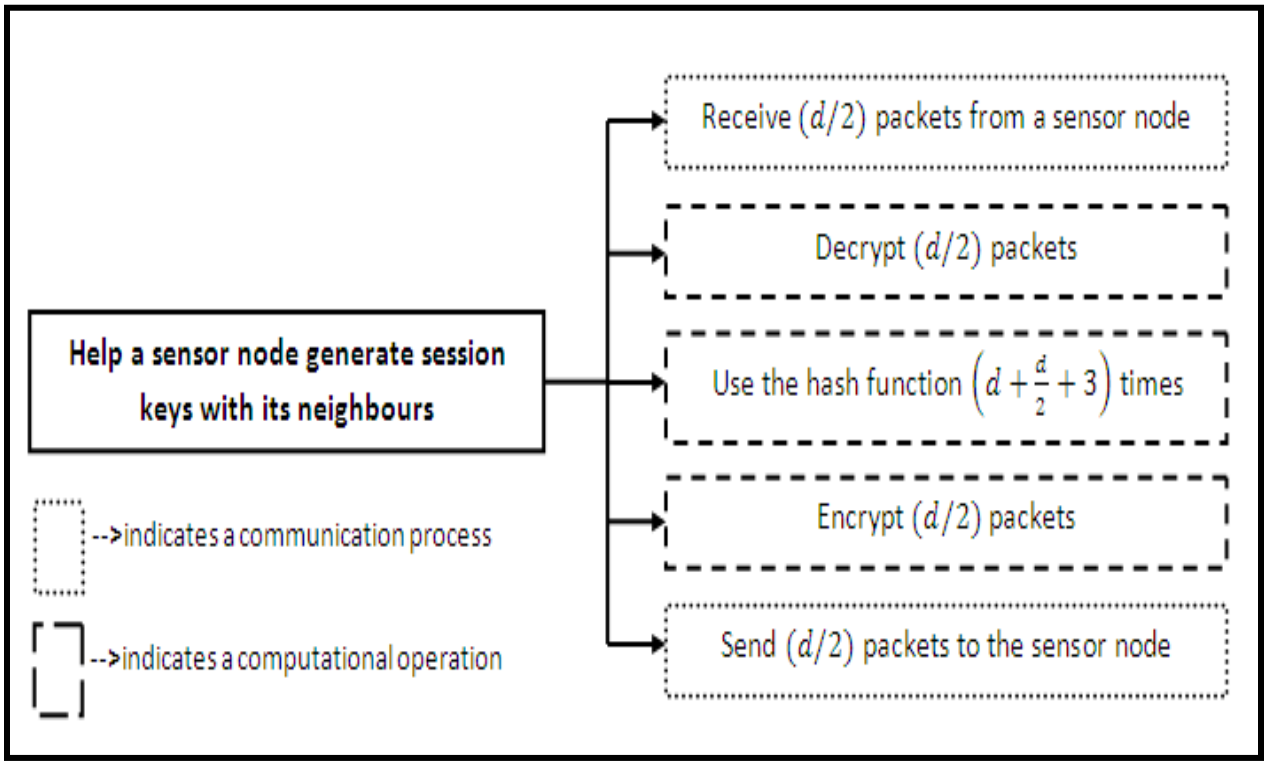

Figure 13: The main operations performed by a third party.

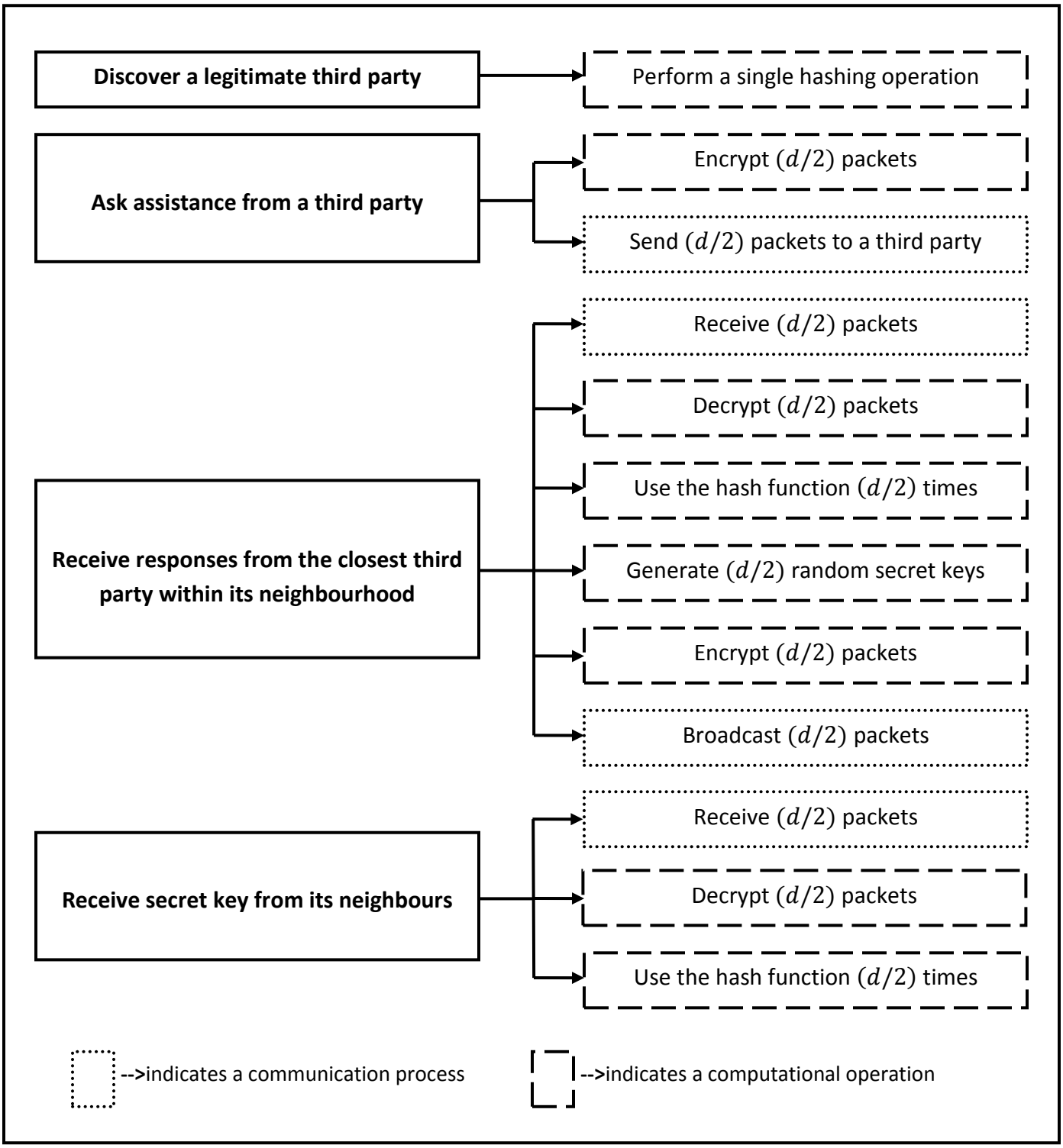

Figure 14: The main operations performed by a sensor node.

3.7 Analysis and Discussion

The proposed method has important advantages compared with the *basic scheme* [27], the random pair-wise key scheme [28], the multiple-space matrix pre-distribution scheme [35], and the multiple-bivariate polynomial pre-distribution scheme [37]. For instance, the proposed scheme not only secures the transmission channels of nodes but also guarantees high local connectivity of the sensor network, low usage of memory resources, and perfect network resilience against node capture. Using a number of third parties equal to 10% of the number of sensor nodes in the field of interest, the proposed scheme achieves 99.42% local connectivity when the expected number of nodes in a neighbourhood is equal to 40, as shown in Figure 7. In this case, sensor nodes need to store only 48 Bytes in their memory. Moreover, capturing a sensor node does not compromise the radio connections of non-compromised nodes, thus providing perfect resilience against node capture.

Different from the proposed method, the key establishment schemes introduced in [27, 28, 35, 37] suffer from a poor trade-off between connectivity, security, and memory usage. For example, it has been shown that the number of symmetric keys stored on a sensor node affects the performance of the *basic scheme* in [27]. To illustrate, increasing the number of secret keys enhances the local connectivity of the sensor network, but negatively impacts the resilience against node capture and obviously increases the memory usage in sensor nodes. Table 3.5 presents the performance of the *basic scheme* for various scenarios obtained by varying the number of secret keys stored on a sensor node. From the table, the key agreement process in [27] consumes 320 Bytes of node memory and results in a low local connectivity of 3.93% if every sensor node is equipped with 20 keys. In such a situation, capturing 11.50% of the sensor nodes will compromise 90% of the secure radio links in the network. When the number of symmetric

keys stored on a sensor node is increased to 100 keys, this process will consume 1600 Bytes of memory but enhance the local connectivity of the network by a factor of around 12 to 47.54%. However, the resilience against node capture in this case is much lower allowing an adversary to compromise 100% of the secure radio connections once 9.86% of the sensor nodes are captured.

The Basic Scheme in [27] when its pool contains 10000 secret keys			
Scenario	Local Connectivity	Memory Usage	Resilience against node capture
1	3.93 %	320 Bytes	Capturing 11.50% of the nodes leads to compromise 90% of the secure links in the network
2	14.48 %	640 Bytes	Capturing 11.50% of the nodes leads to compromise 99% of the secure links in the network
3	30.38 %	960 Bytes	Capturing 7.66% of the nodes leads to compromise 99% of the secure links in the network
4	47.54 %	1280 Bytes	Capturing 5.73% of the nodes leads to compromise 99% of the secure links in the network
5	63.58 %	1600 Bytes	Capturing 9.86% of the nodes leads to compromise 100% of the secure links in the network
6	76.72 %	1920 Bytes	Capturing 8.21% of the nodes leads to compromise 100% of the secure links in the network
7	86.30 %	2240 Bytes	Capturing 7.03% of the nodes leads to compromise 100% of the secure links in the network
8	92.58 %	2560 Bytes	Capturing 6.15% of the nodes leads to compromise 100% of the secure links in the network
9	96.31 %	2880 Bytes	Capturing 5.64% of the nodes leads to compromise 100% of the secure links in the network
10	98.31 %	3200 Bytes	Capturing 4.91% of the nodes leads to compromise 100% of the secure links in the network

Table 3.5: The performance of the basic scheme in [27] for ten scenarios of memory usage.

According to Liu and Ning in [37], the performance of the multiple-space matrix pre-distribution scheme [35] and the multiple-bivariate polynomial pre-distribution scheme [37] are equivalent. Based on this ground, it is sufficient to focus the discussion here on the method in [35]. The local connectivity of the multiple-space scheme as well as its resistance to node capture are considerably influenced by two parameters: ω which denotes the number of private matrices in the network and τ which represents the number of unique vectors chosen randomly for each node. To decrease the local connectivity of a network but improve its resilience against node capture, trusted base stations should increase the number of private matrices in the network and decrease the number of unique vectors chosen for each sensor node. Similar to the *basic scheme*, the number of secret keys stored on a sensor node also plays a significant role in determining the performance of the multiple-space method. Nonetheless, this number influences the memory usage and the resistance of the network to compromised nodes, but it does not affect the local connectivity. As shown in Table 3.6, the key establishment in [35] ensures a high local connectivity of 99.6% and utilizes 1600 Bytes of storage resources if 10 private matrices are available and 5 unique vectors are selected for each node. Yet, the security resilience in this situation is very low since capturing 0.80% of the sensor nodes will break the security of all radio connections in the network. To improve the network resistance to compromised nodes, several solutions can be considered. First, sensor nodes can be equipped with 200 secret keys instead of 100 keys. This option increases the memory usage to 3200 Bytes and slightly improves the network security, as the entire network is now compromised once an adversary captures 1.22% of the sensor nodes. Another option is decreasing the number of unique vectors chosen for the nodes to 3 rather than 5. In this case, the resilience is enhanced as compromising all secure channels requires capturing 1.86% of the sensor nodes, but the local connectivity of

the network drops to 70.83%. A third approach is to increase the number of private matrices from 10 to 90. In this scenario, the network resistance to compromised nodes is improved by a factor of 13, whereas the local connectivity is reduced approximately by a factor of 6 leading to a local connectivity of only 16.89%.

The multiple-space matrix pre-distribution scheme in [35] when n = 10000			
Scenario	Local Connectivity	Memory Usage	Resilience against nodes capture
1	99.6 %	1600 Bytes $w = 10\ and\ \tau = 5$	Capturing 0.80% of the nodes leads to compromise 100% of the secure links in the network
2	99.6 %	3200 Bytes $w = 10\ and\ \tau = 5$	Capturing 1.22% of the nodes leads to compromise 100% of the secure links in the network
3	92.86 %	1600 Bytes $w = 10\ and\ \tau = 4$	Capturing 1.10% of the nodes leads to compromise 100% of the secure links in the network
4	92.86 %	3200 Bytes $w = 10\ and\ \tau = 4$	Capturing 1.88% of the nodes leads to compromise 100% of the secure links in the network
5	70.83 %	1600 Bytes $w = 10\ and\ \tau = 3$	Capturing 1.86% of the nodes leads to compromise 100% of the secure links in the network
6	70.83 %	3200 Bytes $w = 10\ and\ \tau = 3$	Capturing 3.22% of the nodes leads to compromise 100% of the secure links in the network
7	80.63 %	1600 Bytes $w = 20\ and\ \tau = 5$	Capturing 1.47% of the nodes leads to compromise 100% of the secure links in the network
8	62.44 %	1600 Bytes $w = 20\ and\ \tau = 4$	Capturing 2.20% of the nodes leads to compromise 100% of the secure links in the network
9	40.35 %	1600 Bytes $w = 20\ and\ \tau = 3$	Capturing 3.66% of the nodes leads to compromise 100% of the secure links in the network
10	29.14 %	1600 Bytes $w = 50\ and\ \tau = 4$	Capturing 5.88% of the nodes leads to compromise 100% of the secure links in the network
11	16.89 %	1600 Bytes $w = 90\ and\ \tau = 4$	Capturing 10.40% of the nodes leads to compromise 100% of the secure links in the network

Table 3.6: The performance of the scheme in [35] for different values of ω and τ.

The random pair-wise key scheme [28] is designed to ensure perfect resilience against node capture. To accomplish this objective, every cryptographic key in the sensor network is exclusively used to secure a single transmission channel. Consequently, compromising a sensor node reveals only its own key material without affecting the secrecy of the radio connections belonging to other nodes. However, this method also suffers from a trade-off between memory usage and local connectivity. To increase the local connectivity of a sensor network, nodes need to consume more of their available storage resources. Table 3.7 evaluates the random pair-wise key scheme by indicating the number of secret keys stored on a node and the corresponding local connectivity.

The random pair-wise key scheme in [28] when n = 10000			
Scenario	Local Connectivity	Number of Secret Keys	Memory Usage
1	0.10 %	10 keys	160 Bytes
2	0.20 %	20 keys	320 Bytes
3	0.50 %	50 keys	800 Bytes
4	1.00 %	100 keys	1600 Bytes
5	2.00 %	200 keys	3200 Bytes
6	5.00 %	500 keys	8000 Bytes
7	7.00 %	700 keys	11200 Bytes
8	9.00 %	900 keys	14400 Bytes
9	10.00 %	1000 keys	16000 Bytes

Table 3.7: The performance of the pair-wise scheme in [28] for 9 scenarios of memory usage.

As mentioned in Chapter 2, Dong and Liu introduced a key establishment scheme [47] which is similar to our proposed method from the perspective that both techniques employ the concept of third parties in wireless sensor networks. Comparing the two schemes, we indicate that the local connectivity of both methods should be identical given that third parties as well as sensor nodes are uniformly deployed in the field of interest. However, they differ because the expressions used to determine the local connectivity in [47] are only estimates. This is clearly shown in Figure 15. Using a number of third parties equal to 10% of the total number of sensor

nodes, Dong and Liu approximate that one of two neighbouring nodes can discover a third party within its transmission range with a probability of 83.79%. This estimated result is almost 9 percent less than the actual value calculated using (23). Therefore, it is recommended that (23), (25), and (27) be used in order to obtain a precise calculation for the local connectivity of both schemes. Regarding the memory usage, every sensor node using the key agreement scheme in [47] needs to store a single secret key in its storage space. On the other hand, third parties in [47] need a massive amount of memory for key establishment in the network. Not equipping third parties with large memory units eventually makes the scheme in [47] impractical for large networks. On the contrary, both sensor nodes and third parties in the proposed method utilize a small number of secret keys in their storage space. As a consequence, the proposed scheme outperforms that in [47] with regards to memory usage. Assuming the number of sensor nodes is 10000 nodes, Table 3.8 illustrates the memory usage for both schemes.

The Node Type	Dong and Liu Scheme in [47]	Our Proposed Scheme
Sensor Node	A single key = 128 bits	Three keys = 384 bits
Third Party	10000 keys = 160,000 Bytes	Four keys = 512 bits

Table 3.8: Comparing the memory usage of the proposed scheme with the method in [47].

The Dong and Liu scheme [47] is quite resistance to node capture for two reasons: only public information is stored on third parties and private keys are known only to the corresponding sensor nodes. Still, capturing a third party may result in compromising the radio connections of nodes whose session keys were generated by that third party. Once a third party is captured, the adversary can obtain its random number generator and then retrieve the session keys previously generated. To evaluate the network resilience of the method in [47], Figures 16 and 17 show the percentage of compromised transmission channels for a given number of compromised nodes. Both figures show that the ratio of the number of third parties to the number

of sensor nodes has a significant impact on the network resilience against node capture. A ratio greater than 10% will result in good resistance to compromised nodes, especially in sensor networks whose node density is high. For example, at a ratio equal to 10%, capturing 70% of the nodes leads to only 25% of the secure radio connections being compromised in the sensor network when the expected number of nodes in a neighbourhood is equal to 40. Decreasing this ratio from 10% to 1% may provide unsatisfactory performance. In this situation, capturing 10% of the nodes results in 63% and 40% of the secure radio channels being compromised when the node density is 20 and 40, respectively. Different from the key agreement scheme in [47], our proposed scheme ensures perfect resilience against node capture. To achieve this advantage, both sensor nodes and third parties avoid storing private information in their memory. Also, every sensor node is responsible for generating the pair-wise keys needed to secure the transmission channels with its neighbours. Following these two measures confines the negative impact of a compromised sensor node to the node itself.

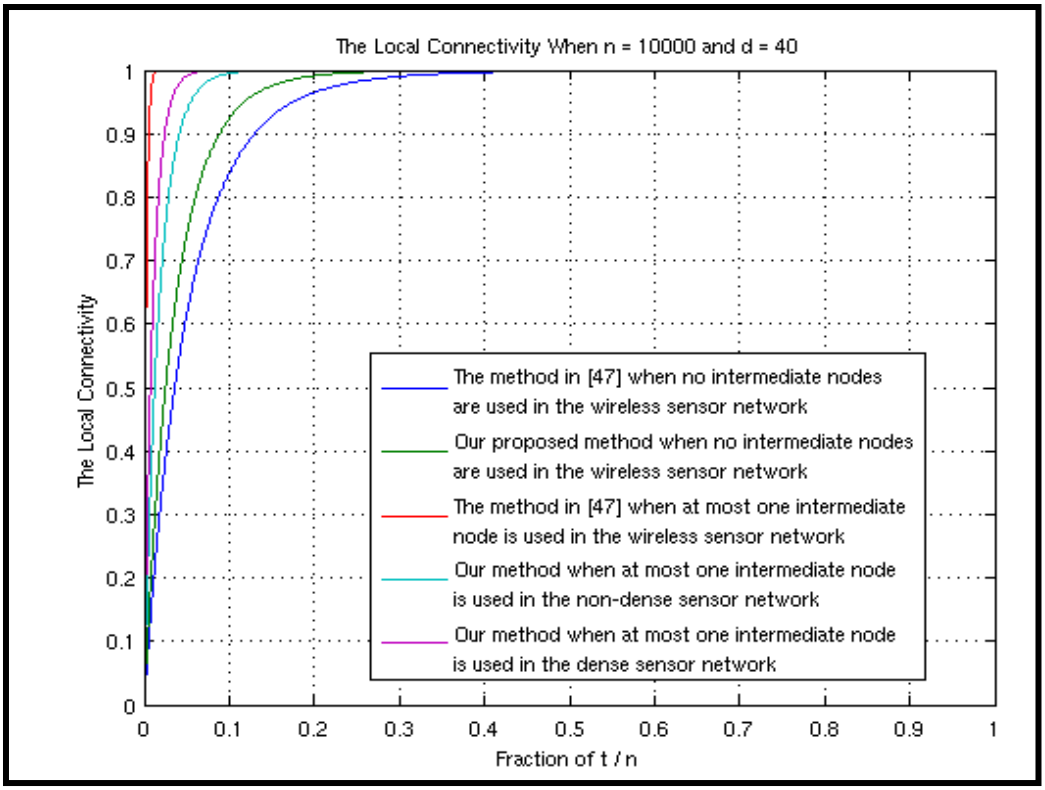

Figure 15: Comparing the local connectivity of the proposed scheme with the method in [47].

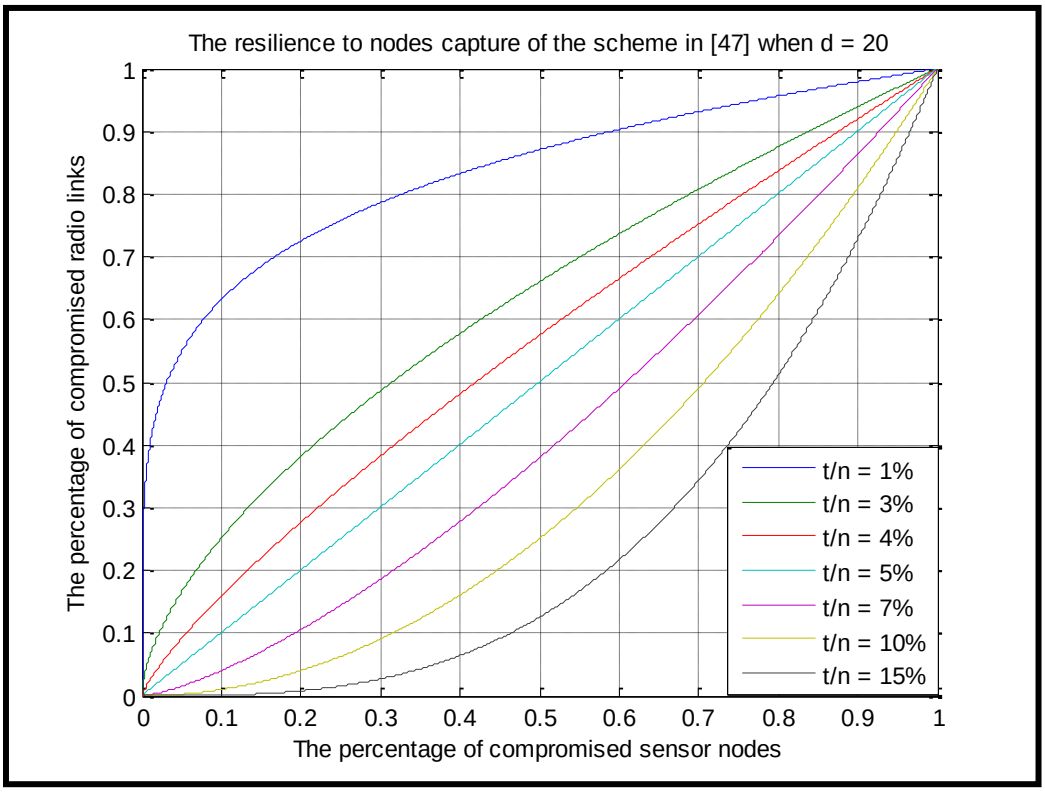

Figure 16: The network resilience of the scheme in [47] when d = 20.

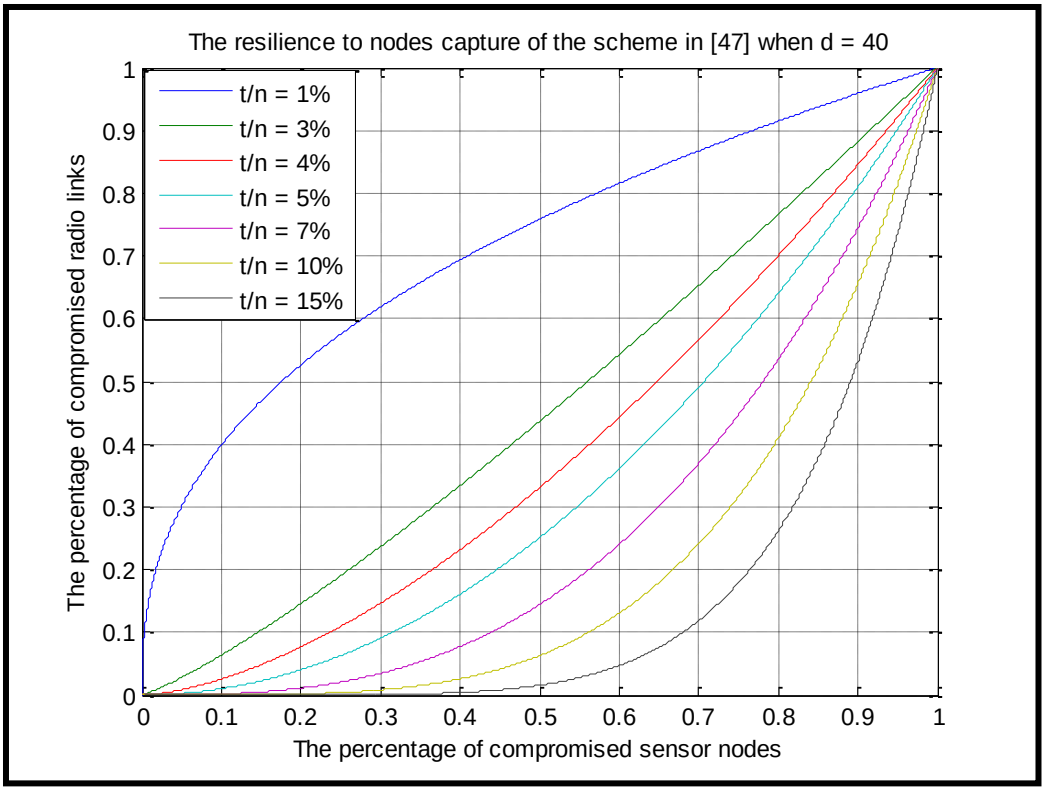

Figure 17: The network resilience of the scheme in [47] when d = 40.

Chapter 4

Key Establishment Using Third Parties in the Presence of Deployment Knowledge

4.1 Introduction

In the previous chapter, sensor nodes and third parties were uniformly deployed in the field of interest, so these units can reside in any portion of the network with equal probability. Nonetheless, the method of deployment in most applications of wireless sensor networks can be exploited to predict the location of a node. To illustrate, a trusted base station typically arranges sensor nodes and third parties into a number of groups prior to the deployment phase. These groups are distributed in the target region utilizing a means of transportation such as trucks or helicopters. Normally, groups of units are dropped one after another while the vehicle is passing through the field of interest. As a consequence, it is possible based on the deployment method to know not only the approximate location of any group of sensor nodes or third parties but also the identities of units residing in their neighbouring groups. According to Du et al. [33], such knowledge can be used to improve the local connectivity of random pre-distribution schemes, strengthen resilience against node capture, and reduce the node memory required to store secret keys. The reason behinds this enhancement is that sensor nodes in random pre-distribution schemes have a greater chance to communicate with nodes in their own groups and neighbouring groups than with nodes residing in distant groups. Therefore, random pre-distribution with deployment knowledge can be redesigned to ensure that sensor nodes only share common secret

keys with nearby nodes. This reduces the number of keys that need to be stored and the overhead for multi-hop communication.

Taking advantage of deployment knowledge, this chapter presents two novel key agreement schemes in which third parties are utilized to secure the transmission channels of nodes in wireless sensor networks. In both methods, a trusted base station divides the sensor nodes into a number of groups and then each group is deployed at a specific location in the field of interest. If at least one third party is assigned to every group, sensor nodes in the network will be capable of establishing pair-wise keys with nodes in their neighbourhoods. In order to describe the proposed schemes in detail, we start this chapter by introducing a simple model for node deployment based on the Gaussian distribution. After that, both methods will be discussed with regards to three important operations: distribution of secret shares to sensor nodes, discovery of local nodes in a neighbourhood, and creation of secure radio channels in the network. To conclude this chapter, the performance of the two schemes is analyzed with respect to local connectivity and memory usage.

4.2 The Deployment Knowledge Model and the Network Configuration

Before modeling the deployment knowledge for the proposed methods, it is necessary to define two important terms: the deployment point and the resident point [33]. Simply, the deployment point is the desired location at which a sensor node or a third party is intended to be placed. Prior to the deployment phase, trusted base stations determine this point for each sensor node and third party. The resident point is the actual location where a sensor node or a third party resides after they are scattered in the field of interest. Even though the resident point is unlikely to be identical to the deployment point, the distance between both points usually follows a

certain probability density function. This probability density function depends on the method of distributing nodes in the wireless sensor network. Similar to the deployment model in [33], we choose the normal (Gaussian) distribution to model the deployment for our proposed schemes. Using a Gaussian distribution with standard deviation σ, it was shown in [33] that the distance between the resident point and the deployment point is smaller than 3σ with a probability equal to 99.87%. Since the proposed schemes deploy every group of sensor nodes and third parties at a unique deployment point, there is little chance that a pair of sensor nodes in two different groups can communicate with each other when the distance between the deployment points of their belonging groups is greater than 6σ.

Figure 18 shows the deployment knowledge model applied to the proposed schemes. In this model, the field of interest is divided into a number of non-overlapping cells. For instance, assuming the network is a two-dimensional area, the target region is partitioned into cells $C(i,j)$ where $i = \{1, 2, \dots, a\}$ and $j = \{1, 2, \dots, b\}$. Normally, the shape of these cells can be either a square [33] or a hexagon [54]. Because sensor nodes and third parties are equipped with omni-directional antennas, a data packet is typically broadcast with circular transmission coverage. Hence, employing hexagonal cells instead of square ones in the deployment model will give a better approximation for the neighbourhood around a deployment point. Prior to the deployment phase, a trusted base station arranges the sensor nodes into $(a \times b)$ groups. Each group consists of d sensor nodes and a single third party. Once these groups are scattered in the target region, group $G_{i,j}$ is expected to be located at the centre of the hexagonal cell $C(i,j)$, which is considered to be its deployment point. From Figure 18, the distance between two neighbouring deployment points is L. Since the resident point of a node k in the group $G_{i,j}$ follows a two-

dimensional Gaussian distribution, the probability density function for this node according to [33] is given by

$$f_k(x, y|k \in G_{i,j}) = \frac{1}{2\pi\sigma^2} e^{-\frac{[(x-x_i)^2+(y-y_i)^2]}{2\sigma^2}} \qquad (28)$$

where the point (x_i, y_j) represents the deployment point for group $G_{i,j}$.

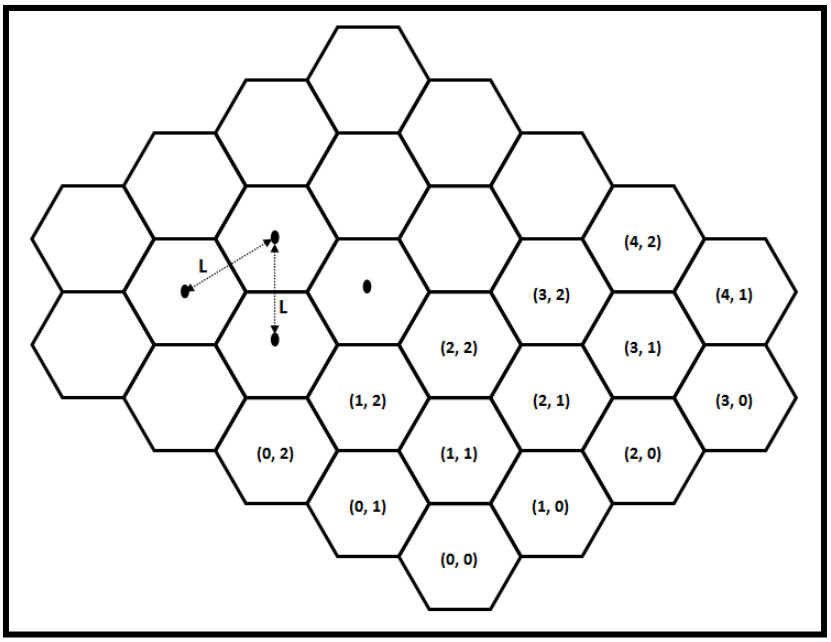

Figure 18: A network in which the distance between adjacent deployment points is equal to L.

The standard deviation of the probability density function in (28) plays a significant role in the local connectivity of a sensor network, as well as the global connectivity. For example, making the distance between the deployment points of two adjacent groups larger than 6σ helps to ensure that sensor nodes and third parties reside in their corresponding cells. In such situations, sensor nodes will usually communicate with their expected neighbours with which they share some secret keys. As a result, the local connectivity within the network cells is considerably enhanced. On the contrary, there will be a low number of sensor nodes at the boundaries between neighbouring cells leading to possible isolation for some groups in the field of interest. Clearly, this isolation will cause a substantial decrease in the global connectivity of

the network. In order to provide good coverage for the wireless sensor network and guarantee an acceptable global connectivity, trusted base stations should distribute the sensor nodes evenly in the area of interest, even in cases when a non-uniform deployment model is used. To find the value of the standard deviation at which sensor nodes are somewhat evenly deployed in the network, we plot the average deployment distribution of sensor nodes over the entire network for various values of σ in Figures 19 to 21. If a node can belong to any particular group with equal probability, the deployment distribution can be represented as follows [33]

$$f_{overall}(x,y) = \sum_{i=1}^{a}\sum_{j=1}^{b}\frac{1}{a \times b} f_k\left(x,y \middle| k \in G_{i,j}\right) \qquad (29)$$

From Figure 21, it is noticeable that using a Gaussian distribution with standard deviation $\sigma = 40m$ provides a nearly even deployment in the sensor network except in small areas near the boundaries of the hexagonal cells. Thus, this value $\sigma = 40$ will be used to deploy the sensor nodes over the target region. Different from sensor nodes, the single third party in the group $G_{i,j}$ should be located more accurately at the deployment point of its group. Actually, placing a third party close to its deployment point (x_i, y_i) facilitates the process of establishing secure radio connections in the sensor network without impacting coverage. For this reason, the standard deviation utilized to deploy third parties should be made small such that accurate deployment is ensured. During our analysis and simulation in the next sections, we will apply the following network configuration:

- The number of sensor nodes in the network is $n = 10000$.
- Sensor nodes and third parties are spread in a region whose area is $800m \times 800m$.
- The average number of nodes within transmission range is either $d = 20$ or $d = 40$.
- The field of interest is divided into (n/d) non-overlapping hexagonal cells.

- Each hexagonal cell is inscribed into a circle whose diameter is $Dim = 80m$.
- Sensor nodes and third parties are deployed at their deployment points using a two-dimensional Gaussian distribution whose standard deviations are $\sigma_{sensor} = 40m$ and $\sigma_{party} = 3m$, respectively.
- The wireless communication range for each node and third party is $R = 40m$.

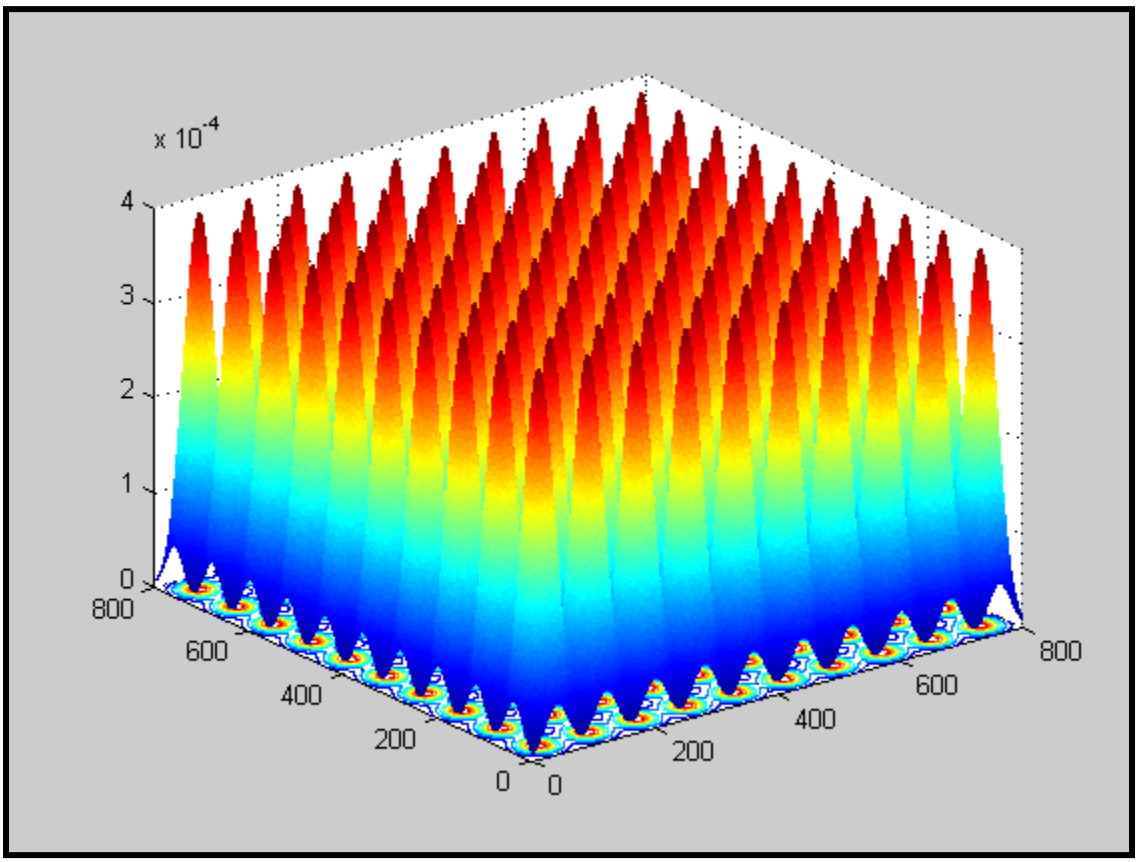

Figure 19: The deployment distribution of sensor nodes when the standard deviation is 20.

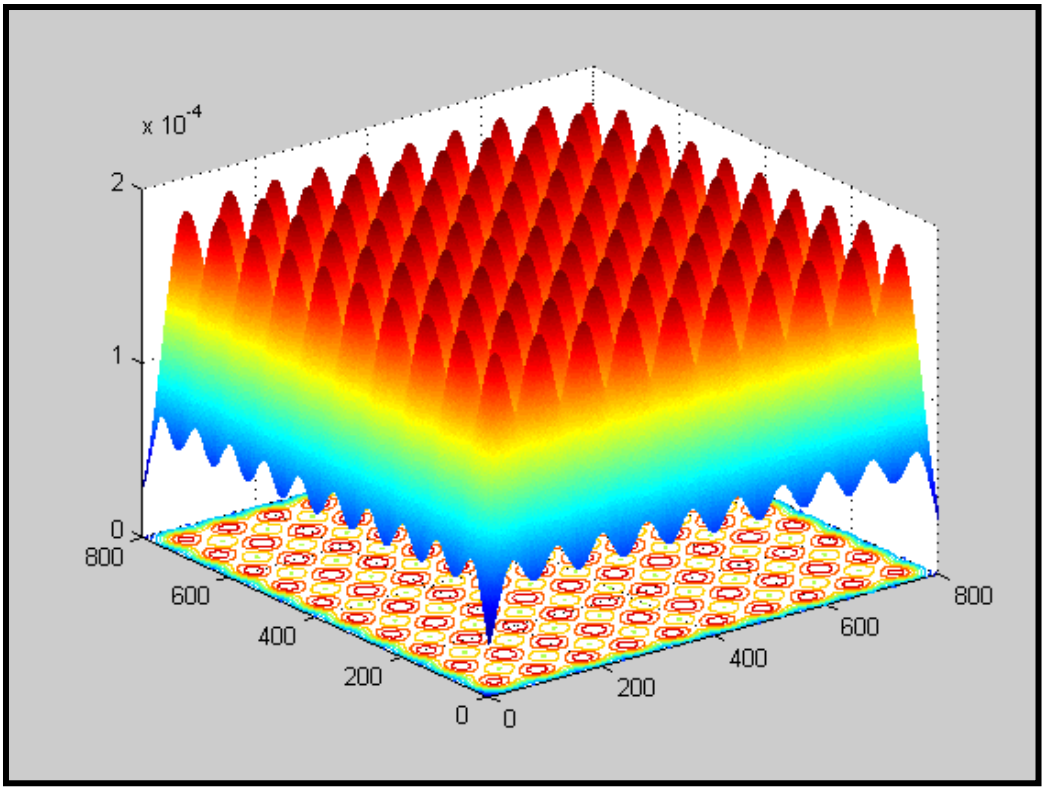

Figure 20: The deployment distribution of sensor nodes when the standard deviation is 30.

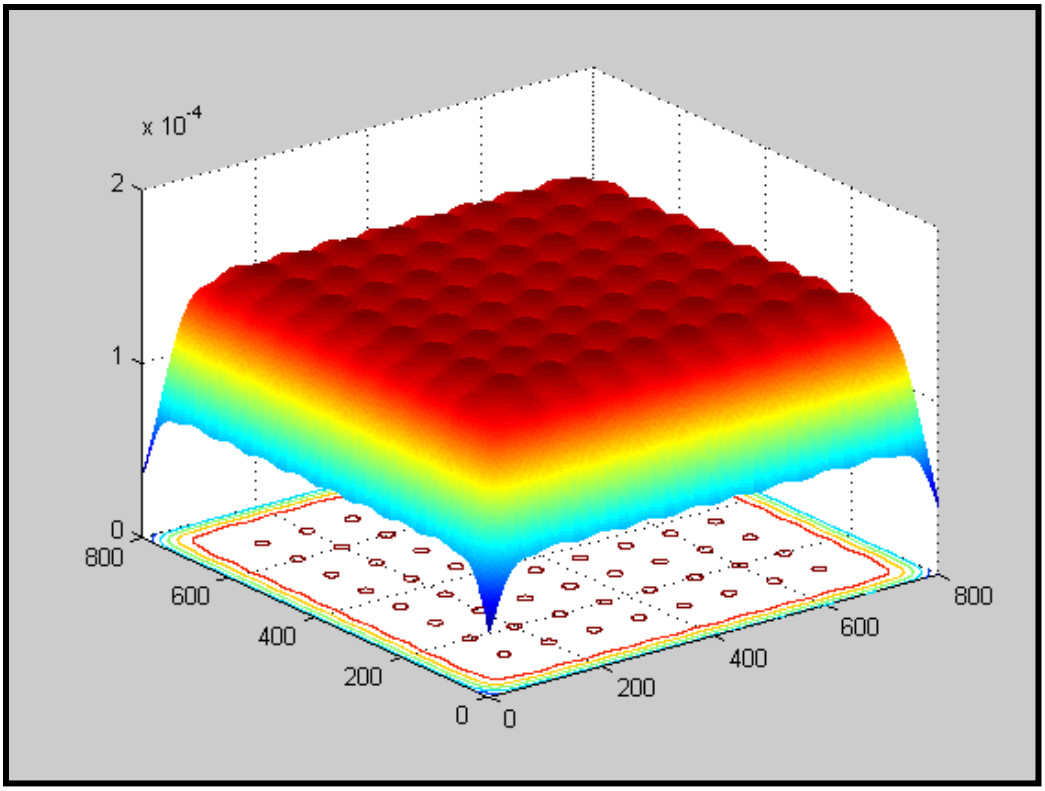

Figure 21: The deployment distribution of sensor nodes when the standard deviation is 40.

4.3 Key Agreement in the Presence of Deployment Knowledge

Using the deployment knowledge model introduced in Section 4.2, we present two key agreement schemes in which sensor nodes create secure radio connections with their neighbours depending on the assistance of third parties. In both methods, a trusted base station divides the sensor nodes into a number of groups and then assigns a single third party to each one of them. Next, every group is placed at a specific deployment point in the field of interest. In order to generate pair-wise keys among nodes in a neighbourhood, in both key agreement schemes three essential steps need to be accomplished: distributing secret shares, discovering local neighbours, and establishing secure channels. We next describe the first method which can be considered as an extension of the scheme proposed in Chapter 3. After that, we explain the second method which is an improved version of the scheme by Dong and Liu [47].

4.3.1 Applying the Deployment Knowledge on the Proposed Method in Chapter 3

It was demonstrated that the method proposed in the previous chapter guarantees low memory usage in the sensor nodes and high local connectivity for the entire sensor network. Moreover, the proposed scheme ensures perfect network resilience against node capture under the assumption that adversaries require time longer than T_k seconds to capture a third party in the field of interest. In this subsection, we apply our deployment knowledge model to the proposed method to increase its flexibility. Indeed, we find that having deployment knowledge assists in making third parties in the network more independent. In other words, each third party in the extended scheme can have unique private information, thereby confining the negative impact of compromising a third party to its local and perhaps nearby neighbourhoods. In addition, the proposed method is generalized by suggesting various solutions such that the assumption made in Chapter 3 can be eliminated.

Before the deployment of a sensor network, a trusted base station in the extended scheme should divide the sensor nodes into a number of groups in which each group consists of d sensor nodes and a single third party. For every group $G_{i,j}$, the trusted base station generates a random encryption key $S_{i,j}$ as well as a random authentication key $A_{i,j}$. Subsequently, it preloads these two secret keys as private information into the memory of the third party belonging to that group. Furthermore, each sensor node in the network is equipped with exclusive encryption and authentication keys in order to communicate with nearby third parties. Since our deployment knowledge model follows a Gaussian distribution whose standard deviation is σ, sensor nodes are expected to be located within distance 3σ of their resident point with probability 99.87%. Therefore, sensor nodes in the group $G_{i,j}$ are likely to interact with third parties residing in the six neighbouring cells in addition to the one located in their own group, as shown in Figure 22. Based on this, the trusted base station should compute the secret keys in Table 4.1 for each sensor node z in the group $G_{i,j}$ and then preload these 14 keys into node memory.

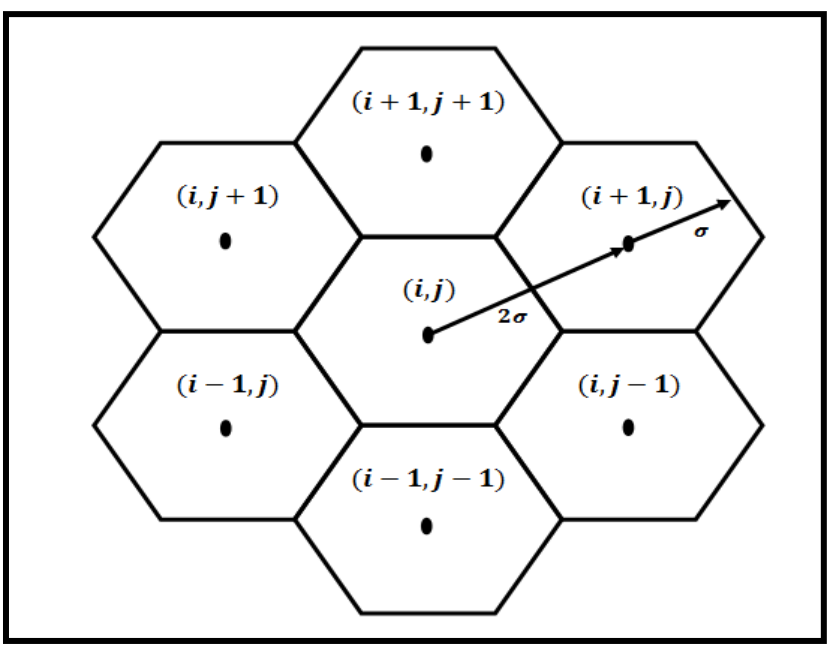

Figure 22: The neighbouring cells of sensor nodes in the group $G_{i,j}$.

Encryption Keys	Authentication Keys
$S_{z,1} = Hash(S_{i,j}, ID_z)$	$A_{z,1} = Hash(A_{i,j}, ID_z)$
$S_{z,2} = Hash(S_{i-1,j-1}, ID_z)$	$A_{z,2} = Hash(A_{i-1,j-1}, ID_z)$
$S_{z,3} = Hash(S_{i,j-1}, ID_z)$	$A_{z,3} = Hash(A_{i,j-1}, ID_z)$
$S_{z,4} = Hash(S_{i-1,j}, ID_z)$	$A_{z,4} = Hash(A_{i-1,j}, ID_z)$
$S_{z,5} = Hash(S_{i,j+1}, ID_z)$	$A_{z,5} = Hash(A_{i,j+1}, ID_z)$
$S_{z,6} = Hash(S_{i+1,j}, ID_z)$	$A_{z,6} = Hash(A_{i+1,j}, ID_z)$
$S_{z,7} = Hash(S_{i+1,j+1}, ID_z)$	$A_{z,7} = Hash(A_{i+1,j+1}, ID_z)$

Table 4.1: The secret keys stored into the memory unit of node z in the group $G_{i,j}$.

After scattering sensor nodes in the target region, they perform a discovery process, similar to that explained in Section 3.2, such that their local neighbours as well as their nearby third parties are found. Even though either the white list approach or the hash chain method can be utilized to verify the legitimacy of a third party in a neighbourhood, we prefer to use a white list in the extended method. Due to the availability of deployment knowledge, sensor nodes need to store the identities of only seven third parties, thus making the white list mechanism feasible to be implemented in this key agreement scheme. If the sensor network distributes additional third parties to substitute for failed ones, the trusted base station will broadcast encrypted messages to all affected groups stating the identities of both the revoked and new third parties in their regions. Unlike with the scheme proposed in the previous chapter, using the white list method instead of the hash chain mechanism does not consume a substantial amount of node memory resources. The total power consumption in the network is reduced because sensor nodes and third parties do not require the execution of hash functions or the transmission of customized Hello packets during verification.

When the discovery stage is completed, nodes next establish secure radio connections with their neighbours. To accomplish this objective, every pair of adjacent nodes must agree on a common third party whose identity is stored in the predefined white lists of both sensor nodes.

After that, one of the two nodes will contact the chosen third party. Selecting which one of the two is responsible for communicating with the third party can depend on two parameters: the remaining power level on both nodes and their proximity to the common third party. As soon as this decision is made, the responsible node will follow the steps in Algorithm 4.1 so that a pair-wise key is generated. Since it is possible to encounter situations in which two neighbouring nodes have no third parties in common, it is necessary to include an additional step in the key establishment phase to deal with this situation. In this step, a common neighbour of the two nodes will help them in finding a third party within a predefined number of hops and then communicate with that third party on behalf of the two nodes. Upon receiving the anticipated response from the third party, the common neighbour will forward the incoming response to one of the two nodes. This node will continue the process of key establishment in Algorithm 4.1 in order to initiate a secure radio channel with its neighbouring node.

If a sensor node or third party in a certain group is captured, the proposed key agreement scheme ensures in most situations that there is no impact on the secrecy of the transmission channels belonging to other units in the network. Nonetheless, a compromised third party may still obstruct future key establishments within its local and nearby groups. To clarify, it is possible to find pairs of neighbouring nodes which have no common third parties except the compromised one. In this scenario, each pair of these nodes should rely on an intermediate node (within one hop) to perform the key establishment process. If a common intermediate node exists, a secure radio connection is created between the pair of neighbouring nodes. This does, however, increase the communication overhead of the key agreement scheme. Otherwise, there will be no pair-wise key generated for the two nodes, resulting in a decrease in the local connectivity of the wireless sensor network.

Algorithm 4.1: Generating a pair-wise key between two neighbouring nodes

1. One of the two sensor nodes sends a request to the common third party $T_{i,j}$ in its transmission range stating the desire to establish a secure radio channel with a neighbouring node z. To help the third party in verifying the identity of the packet originator, node y encrypts its outgoing message with the authentication secret $A_{y,1}$

$$Node(y) \rightarrow Third\ Party: \quad E_{A_{y,1}}\big(Request(y,z)\big)$$

2. The third party begins the process of pair-wise key generation by computing the values $\mathrm{S_{y,1}}$, $\mathrm{A_{y,1}}$, and $\mathrm{S_{z,1}}$. Then it calculates a temporary secret share for use in generating the session key between node y and its neighbour z as follows

$$\mathrm{Secret(y,z) = Hash\big(S_{y,1}, ID_z\big) \oplus Hash\big(S_{z,1}, ID_y\big)}$$

3. The third party sends back to the sensor node y the temporary secret share protected with the authentication key $\mathrm{A_{y,1}}$ as shown below

$$\mathrm{Third\ Party \rightarrow Node(y): \quad E_{A_{y,1}}\big(y \,||\, z \,||\, Secret(y,z)\big)}$$

4. Taking advantage of the identity of its neighbouring node z, the sensor node y determines the session key between both nodes given by

$$\mathrm{Session_z = Secret(y,z) \oplus Hash\big(S_{y,1}, ID_z\big)}$$

5. The sensor node y generates a new random secret key $K_{y,z}$ and then constructs a packet containing this random key. This packet should be encrypted with the session key obtained in the previous step and then forwarded to the neighbouring node z.
6. After receiving the encrypted packet, the neighbouring node z inputs the identity of node y into the predefined hash function in order to extract the session key used to encrypt the received packet as follows

$$Session_y = Hash\big(S_{z,1}, ID_y\big)$$

Next, it uses this session key to decrypt the received packet and then obtains the secret key $K_{y,z}$. At this point, both sensor nodes y and z can employ $K_{y,z}$ to secure their radio channel.

To overcome the problems described above, the key agreement method in this subsection can be modified using one of three countermeasure procedures once a third party is captured. The first procedure deploys a new third party in the affected region to replace the compromised one. At the same time, a trusted base station sends the corresponding encryption and authentication keys to every sensor node residing in the affected region and its neighbouring groups. Furthermore, the white list on these nodes should be updated with the identity of the new third party. As the expected number of nodes in a cell is not small, especially in dense sensor networks, this solution still suffers from high communication overhead due to the necessity of transmitting a large number of update packets to the nodes. Conversely, the second procedure exploits the idea of pairing based cryptography in which sensor nodes exchange a session key without any communication. If two adjacent nodes fail in discovering a common and trustworthy third party, both nodes can switch to an alternative method by applying the non-interactive identity-base scheme in [46], which was explained in Section 2.7. Following this procedure guarantees prefect local connectivity, but it increases the memory usage of our scheme since each sensor node must be equipped with additional security credentials, including an exclusive private key and a hashing-and-mapping function. Although the computational overhead of the scheme in [46] is not negligible, the second procedure is still a reasonable solution. This is because the non-interactive identity-based scheme will rarely be used just to extract the session key required to exchange a symmetric key between two neighbouring nodes. Different from the previous two solutions, the third procedure does not create any burden on the resources of a sensor node including its memory usage, computational processing, and power to transmit data packets. Similar to what was done in Section 3.4, the third procedure assumes that an attacker requires T_k seconds in order to compromise a third party node. Consequently, deleting the

private information stored on third parties T_k seconds into the network lifetime provides perfect resilience against node capture. When sensor nodes in a certain group require further key establishments, nearby third parties receive from a trusted base station encrypted packets containing the corresponding private information. Again, these neighbouring third parties must delete their secret keys as soon as the key agreement process is completed. When a third party is captured, a substitute third party with the same private information will be deployed in the affected group. During our analysis and discussion in subsequent subsections, this last procedure is used as the countermeasure to compromised third parties.

4.3.2 A Flexible Key Agreement Scheme Using Deployment Knowledge

To provide suitable substitution procedure for captured third parties, it is better to avoid imposing specific assumptions or consuming significant node resources. In this subsection, we present a flexible key establishment scheme for the replacement of a compromised third party in a particular region. The proposed scheme is an improved version of the key agreement method in [47] in that it provides better network scalability and memory usage in the third parties. The reason for this is the availability of deployment knowledge which helps in overcoming the disadvantages of the Dong and Liu method [47].

Before distributing sensor nodes in the field of interest, a trusted base station generates a random secret key K_y for each sensor node y and then preloads this key into node memory. Using deployment knowledge, the base station also predicts the identities of third parties which are deployed within the neighbouring cells of the sensor node y. For every adjacent third party t_i, the base station performs the hash function $H(K_y, ID_{t_i})$ and stores the result and the identity of the sensor node y in the storage space of that third party. Once the sensor nodes are scattered in the target region, they discover their local neighbours and adjacent third parties. As in the

previous subsection, the white list mechanism is used by the sensor nodes to determine the legitimacy of a third party. After completing the discovery phase, two neighbouring nodes y and z can being the process of securing their radio connections. To accomplish this, both sensor nodes must agree on a common third party. Next, one of the two sends a request to this third party asking for assistance in key establishment. To authenticate its identity, the sensor node encrypts the outgoing packet with the predefined key, which is $H(K_y, ID_{t_i})$ if node y is responsible for interaction with the common third party t_i. After the incoming message is decrypted with the key $H(K_y, ID_{t_i})$, the third party uses a cryptographic pseudo-random number generator to generate an exclusive pair-wise key $K_{y,z}$. Then the third party constructs two packets containing this secret key. The first packet will be protected with $H(K_y, ID_{t_i})$ and sent to node y, whereas the other message will be protected with $H(K_z, ID_{t_i})$ and forwarded to the neighbouring node z. At this point, both nodes can use the received pair-wise key $K_{y,z}$ to secure their transmission channel. If a pair of neighbouring nodes does not share a common third party, they must depend on an intermediate node which can reach their common third party within a number of hops. This intermediate node will contact the common third party and resend the incoming response to the two nodes.

When an adversary captures a third party in a certain group, the trusted base station tries to prevent key agreement disturbance by redeploying a new third party (t_{new}) in the affected group. Prior to the deployment of this new third party, a trusted base station knows the identities of the sensor nodes residing in the affected group or at one of its six neighbouring cells. For every such node u, the base station computes the hash function $H(K_u, ID_{t_{new}})$ and then stores the result with the node identity in the memory of the new third party. The white list of the sensor node u is also updated to include the identity of the new third party. Obviously, the

countermeasure procedure used here to replace a compromised third party is not dependent on any assumptions and does not create a significant burden on the resources of the sensor nodes.

4.4 Local Connectivity

In both methods proposed in this chapter, a pair of neighbouring sensor nodes establishes a secure radio connection if the two nodes have a common third party within their transmission range. Based on this definition, the local connectivity of the sensor network is determined using the conditional probability given by

$$p_{local} = Pr\big(B(y,z)|A(y,z)\big) = \frac{Pr\big(B(y,z) \text{ and } A(y,z)\big)}{Pr\big(A(y,z)\big)} \qquad (30)$$

where $A(y,z)$ represents the event that sensor nodes y and z are neighbours, while $B(y,z)$ refers to the case when both nodes share a common third party within their radio coverage. Because sensor nodes y and z are chosen in a random manner, it is important to consider all possible pairs of nodes in the network during the process of calculating the local connectivity for a certain scheme. According to Du et al. [33], this can be achieved by computing the average of (30) over the entire field of interest. Assuming φ to be the set of all groups in a sensor network, the local connectivity in (30) can be rewritten as follows

$$p_{local} = \frac{\sum_{i\epsilon\varphi}\sum_{j\epsilon\varphi} Pr\big(B(y,z) and\ A(y,z) \mid y \epsilon G_i\ and\ z \epsilon G_j\big) \times Pr\big(y \epsilon G_i\ and\ z \epsilon G_j\big)}{\sum_{i\epsilon\varphi}\sum_{j\epsilon\varphi} Pr\big(A(y,z) \mid y \epsilon G_i\ and\ z \epsilon G_j\big) \times Pr\big(y \epsilon G_i\ and\ z \epsilon G_j\big)} \qquad (31)$$

Since $B(y,z)$, $A(y,z)$, $(y \epsilon G_i)$, and $\big(z \epsilon G_j\big)$ are independent events, (31) can be simplified to

$$p_{local} = \frac{Pr\big(y \epsilon G_i\ and\ z \epsilon G_j\big)}{Pr\big(y \epsilon G_i\ and\ z \epsilon G_j\big)} \times \frac{\sum_{i\epsilon\varphi}\sum_{j\epsilon\varphi} Pr\big(B(y,z) and\ A(y,z) \mid y \epsilon G_i\ and\ z \epsilon G_j\big)}{\sum_{i\epsilon\varphi}\sum_{j\epsilon\varphi} Pr\big(A(y,z) \mid y \epsilon G_i\ and\ z \epsilon G_j\big)}$$

$$= \frac{\sum_{i\epsilon\varphi}\sum_{j\epsilon\varphi} Pr\big(B(y,z) and\ A(y,z) \mid y \epsilon G_i\ and\ z \epsilon G_j\big)}{\sum_{i\epsilon\varphi}\sum_{j\epsilon\varphi} Pr\big(A(y,z) \mid y \epsilon G_i\ and\ z \epsilon G_j\big)} \qquad (32)$$

Instead of finding a closed-form expression for (32), we provide a simple approximation for the local connectivity of the proposed schemes. In our deployment knowledge model, the field of interest is divided into a number of non-overlapping cells. At the centre of each cell $C(i,j)$, a group of sensor nodes in addition to a single third party is distributed. Although sensor nodes may reside in a cell different from their expected ones, a third party is usually located very close to its deployment point. This is due to the assumption that the distance between the resident and deployment points of a third party follows a two-dimensional Gaussian distribution whose standard deviation is small. Hence, it is essentially guaranteed that every sensor node residing in a cell can reach the third party assigned to that cell. On the other hand, sensor nodes are located in the target region according to a Gaussian distribution whose standard deviation is $\sigma = 40m$. Since the distance between the deployment points of two adjacent cells is $L = 2\sigma$, Figure 23 shows that 68.27% of the sensor nodes will be located within their expected cells. Furthermore, 99.73% of the sensor nodes will most likely reside either in their local cell or one of six neighbouring cells, so only 0.27% of the nodes will be located in non-adjacent cells.

Without loss of generality, we assume that sensor node y from cell $C(i,j)$ in Figure 24 wants to create a secure radio connection with node z in its neighbourhood. To accomplish this objective, both nodes should discover in their transmission range a third party whose identity is included in their white lists. 99.73% of the sensor nodes in $C(i,j)$ belong to the same local group or adjacent ones, so every pair of these sensor nodes must have at least a common third party that can generate an exclusive key between the nodes. Even though 0.27% of the sensor nodes in cell $C(i,j)$ are not being expected to have the identity of the third party assigned to that cell, these sensor nodes may still succeed in establishing secret keys with some neighbours in $C(i,j)$ without the aid of intermediate nodes. If sensor node g belonging to non-adjacent cell G is

located within $C(i,j)$, this node will share a common third party with nodes belonging to cells B or C and residing within $C(i,j)$. When the sensor nodes are deployed in the field of interest, we can say that g shares a common third party with more than 10% of the sensor nodes located within cell $C(i,j)$. As a consequence, the local connectivity of the proposed schemes in this chapter will be greater than 99.73%.

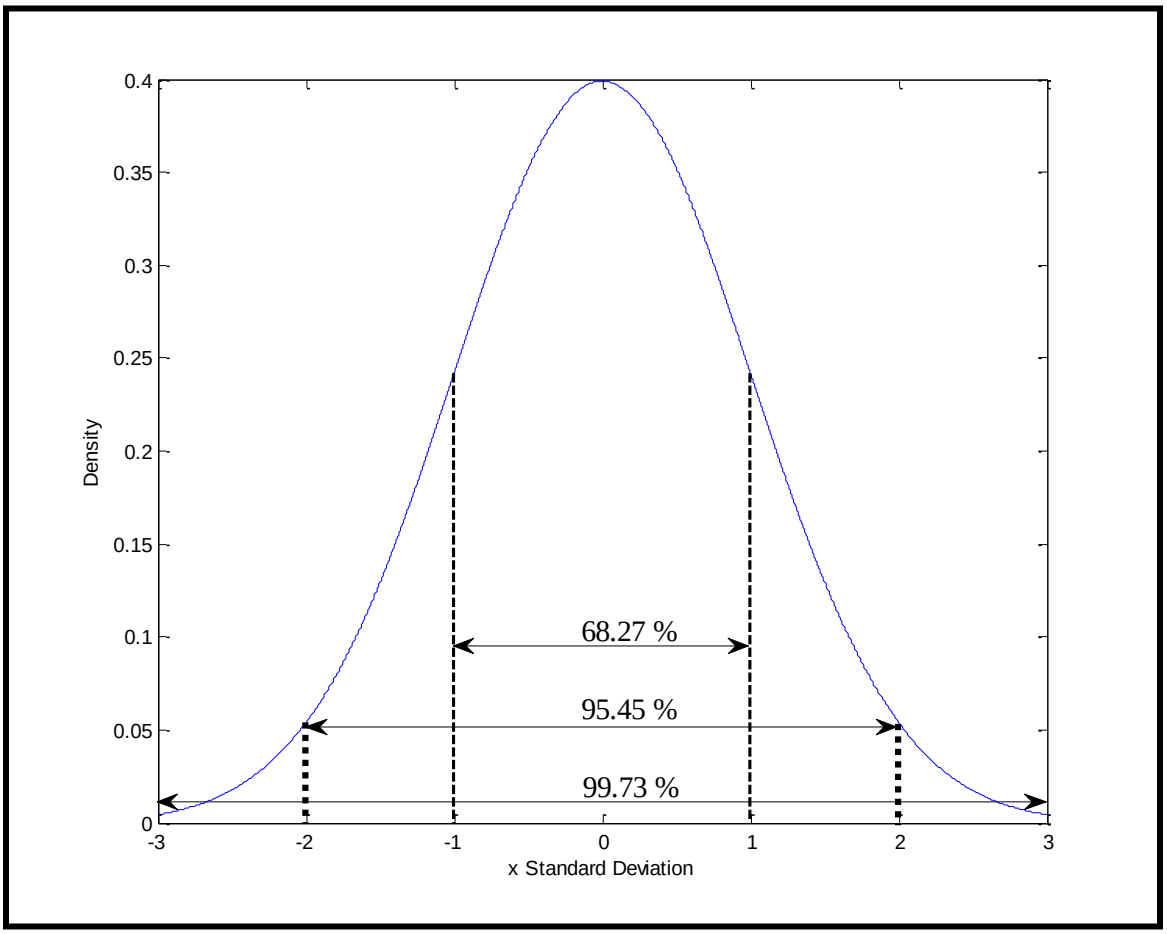

Figure 23: The Gaussian probability density function.

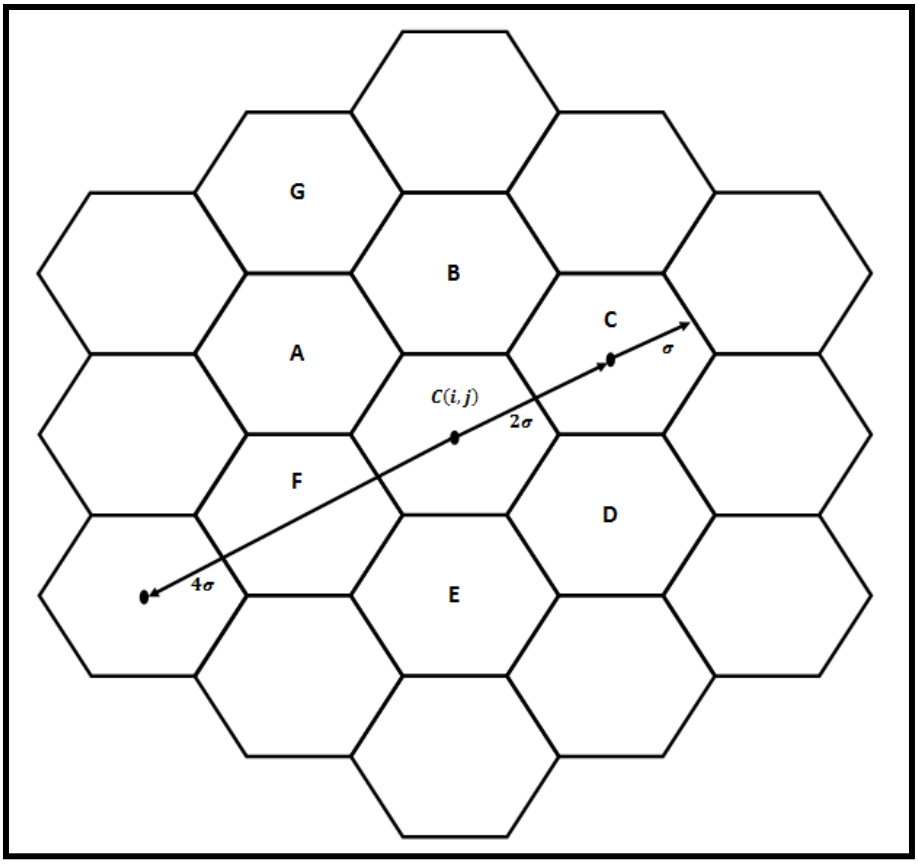

Figure 24: The cells whose nodes may reside within C(i, j).

Both key agreement schemes proposed in this chapter outperform the key establishment method explained in Section 3.2 with regards to local connectivity. Let d be the average number of nodes within a neighbourhood and r be the ratio of the number of third parties to the number of sensor nodes in the network. As illustrated in Section 3.3, the method introduced in the previous chapter provides 74.02% local connectivity when $d = 20$ and $r = 0.05$. On the other hand, the schemes proposed in this chapter provide a local connectivity greater than 99.73% with the same parameters. To achieve the same connectivity, the method in Section 3.2 requires an increase to $r = 0.2632$, which is an infeasible from both practical and economic points of view. Another example for the outstanding performance provided by the schemes proposed in this chapter can be seen in the sensor network with $d = 40$ and $r = 0.025$. In such scenarios, the key establishment schemes in Section 3.2 and Section 4.3 provide a local connectivity of 74.91% and

99.73%, respectively. In order to achieve 99.73% local connectivity, the method in Chapter 3 requires the deployment of almost five times the number of third parties used by the scheme in Section 4.3. Therefore, having knowledge about the network deployment clearly enhances the local connectivity of the key agreement schemes.

4.5 Memory Usage

Unlike the local connectivity metric in which the two key agreement schemes proposed in this chapter provide identical results, the memory requirements to secure the radio connections of a sensor network are different with these methods. For instance, the scheme introduced in Section 4.3.1 needs a small amount of memory in the sensor nodes and third parties. To clarify, every sensor node in the network should store a white list of identities in order to verify the legitimacy of a third party in its neighbourhood. Since sensor nodes are, with high probability, located either within a local cell or one of the six neighbouring cells, the node is preloaded with a white list consisting of at least seven identities. In addition, each sensor node requires fourteen secret keys, i.e. seven pairs of encryption and authentication keys, to communicate with nearby third parties. Regarding the memory usage in third parties, a third party is equipped with three secret keys: an encryption key S to assist a pair of neighbours in generating session keys, an authentication key A to identify the originators of packets sent to or from sensor nodes, and an exclusive pair-wise key B to initiate secure routes to trusted base stations in the network.

The memory usage with the key agreement method presented in Section 4.3.2 requires a considerable amount of storage space in the third parties but little memory in the sensor nodes. Besides the identities of the seven third parties residing within its local and adjacent cells, a sensor node must store a single private key in its memory to secure its transmission channels. On

the other hand, the third party t_i requires storing the hash value $H(K_y, ID_{t_i})$ and the identity of every sensor node y within the cell i or at one of its neighbouring cells. Because the average number of sensor nodes in a certain region is given by d, a trusted base station preloads $7 \times d$ hash images as well as the $7 \times d$ corresponding identities in the storage space of the third party t_i. Assume the node identities and the cryptographic keys are 20 bits and 128 bits in length, respectively. If the expected number of nodes within a transmission range is equal to $d = 20$, the memory usage in the third party t_i is 2590 Bytes. On the ground that third parties are only responsible for the pair-wise key establishment among sensor nodes without conducting other operations such as data sensing or packet routing, storing 140 secret keys on a third party will be a negligible burden on network resources.

As mentioned earlier, the first proposed key agreement scheme in this chapter is basically a modified version of the method presented in Section 3.2. Since the modification was made only in the way of distributing secret keys by taking into account the anticipated locations of the sensor nodes in the field of interest, the performance of the two schemes differs only with respect to the local connectivity and memory usage. In other words, the network resilience against node capture in addition to the communication and computational overhead of both methods will not differ from the results obtained in Sections 3.4 and 3.6, respectively. Table 4.2 provides a comparison between the two methods. From this table, it is obvious that having knowledge about the node deployment enhances the local connectivity of the sensor network with an insignificant increase in the memory usage.

Metric	The Method Introduced in Section 3.2	The Method Proposed in Section 4.3.1
Local Connectivity	74.02% when $d = 20$	99.73% when $d = 20$
	74.91% when $d = 40$	99.73% when $d = 40$
Sensor Node Memory Usage	3 secret keys $\Downarrow$ 384 bits	14 secret keys and 7 identities $\Downarrow$ 1932 bits
Third Party Memory Usage	4 secret keys $\Downarrow$ 512 bits	3 secret keys $\Downarrow$ 384 bits

Table 4.2: Comparison of the method in Section 3.2 with the scheme in Section 4.3.1.

The second key establishment method in this chapter applies the deployment knowledge model to the scheme introduced by Dong and Liu in [47]. When the number of sensor nodes is $n = 10000$ and the network density is $d = 20$, Table 4.3 shows that the deployment knowledge improves both memory usage and local connectivity. As a matter of fact, the method in Section 4.3.2 reduces the memory needed in the third parties by a factor of more than 66.

Metric	Dong and Liu Method [47]	Scheme Proposed in Section 4.3.2
Local Connectivity	74.02%	99.73%
Sensor Node Memory Usage	One secret key $\Downarrow$ 128 bits	One secret key and 7 identities $\Downarrow$ 268 bits
Third Party Memory Usage	10000 secret keys $\Downarrow$ 160,000 Bytes	140 secret keys and 140 identities $\Downarrow$ 2590 Bytes

Table 4.3: Comparison of the method in [47] with the scheme in Section 4.3.2.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis contributes to the challenging field of security for wireless sensor networks by introducing three key agreement schemes in which sensor nodes create secure radio connections with their neighbours depending on the aid of third parties. These third parties are responsible only for the pair-wise key establishment among sensor nodes, so they do not observe the physical phenomenon nor route data packets to other nodes. In the previous chapters, we described each proposed method with respect to four important issues: how secret shares are distributed, how local neighbours are discovered, how legitimate third parties are verified, and how secure channels are established. Moreover, we analyzed the performance of these schemes with regards to five metrics: local connectivity, resistance to node capture, memory usage, communication overhead, and computational burden.

In the first key agreement method, the network uniformly deploys sensor nodes and third parties in the field of interest. In order to secure the radio channel of two neighbouring nodes, one of the two sensor nodes communicates with a third party in its transmission coverage area asking for a session key. Furthermore, this sensor node randomly generates an exclusive secret key and constructs a packet containing this key. The packet is next encrypted by the session key received from the third party and then forwarded to the neighbouring node. In Chapter 3, it was demonstrated that the first scheme does not only secure the radio connections of sensor nodes but also provides high local connectivity for the network, low usage of memory resources, and

perfect network resilience against node capture. Using a number of third parties equal to five percent of the total number of sensor nodes in the area of interest, the first method achieves 93.28% local connectivity if the number of sensor nodes in a neighbourhood is equal to 40 on average. In this case, a sensor node needs to store only 384 bits in its memory. Also, capturing a sensor node or a third party has no negative impact on the radio connections of other non-compromised nodes if the private keys belonging to third parties are deleted within T_k seconds from the beginning of the key agreement phase.

After introducing a model for the deployment knowledge using the Gaussian distribution, we presented the second and the third key agreement schemes in which third parties as well as deployment knowledge are utilized to secure the transmission channels of a wireless sensor network. In both methods, a trusted base station divides the sensor nodes into a number of groups and then assigns a single third party within each of them. Every group is intended to be located at a specific position in the field of interest. To facilitate the process of generating a pair-wise key for two adjacent nodes in a neighbourhood, the second method follows an approach similar to the first proposed scheme. However, we showed that having deployment knowledge enhanced the local connectivity of the network with an insignificant increase in node memory usage. For the second method, three countermeasure procedures were considered to replace a compromised third party, thereby allowing for future key establishment by the sensor nodes. These three procedures differ from each other in terms of power consumption and the necessary assumptions. To provide an economical mechanism for the substitution of captured third parties in the network without imposing any assumptions, we presented a third key agreement scheme whose main characteristic is not allowing third parties to store private keys for the entire network lifetime.

5.2 Future Work

Several possible future research directions can be derived from the work presented in this thesis. For instance, we focused in the previous two chapters on proposing efficient and scalable key agreement schemes which allow a pair of neighbouring nodes to share a unique pair-wise key. Nevertheless, we have not mentioned explicitly any particular procedure to detect and revoke the identities of compromised sensor nodes or third parties in the field of interest. Consequently, it is desirable to provide a simple mechanism that helps in detecting misbehaving nodes and third parties. These captured entities can then be isolated from the sensor network by denying them the ability to exchange messages with authorized units. Clearly, the matter of discovering a compromised entity can be considered an anomaly intrusion detection problem in which suspicious activities are identified. Statistical or economical rules can be used for this purpose instead of depending on specific patterns or signatures. To implement this technique, we suggest training the wireless sensor network to recognize normal activities prior to the network deployment phase. This training can be accomplished with the aid of either artificial intelligence techniques or some predefined mathematical equations. Once sensor nodes and third parties are distributed in the area of interest, each sensor node observes the activities of its neighbours and compares their behaviour with the predicted ones in order to detect abnormal operations. When a compromised entity is identified, the sensor node will deliver its concern to the trusted base stations so that an appropriate action can be performed. Since conventional intrusion detection systems are usually complex and consume a significant amount of node resources, it is important to propose a simple detection and revocation model suitable for the characteristics of wireless sensor networks.

Bibliography

[1] A. Goldsmith, *Wireless Communications*. New York: Cambridge University Press, 2005.

[2] X. Liang, "On Wireless Sensor Networks," M.S. thesis, University of Victoria, Canada, 2008.

[3] A.J. Goldsmith and S.B. Wicker, "Design challenges for energy-constrained ad-hoc wireless networks," *IEEE Wireless Communications*, vol. 9, no. 4, pp. 8–27, August 2002.

[4] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, 2002, pp. 88-97.

[5] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no.2-3, pp. 293-315, September 2003.

[6] C. F. Garcia-Hernandez, P. H. Ibarguengoytia-Gonzalez, and J. A. Perez-Diaz "Wireless sensor networks and applications: A survey," *IJCSNS: International Journal of Computer Science and Network Security*, vol. 7, no. 3, pp. 264-73, 2007.

[7] D. Puccinelli and M. Haenggi, "Wireless sensor networks: Applications and challenges of ubiquitous sensing," *IEEE Circuits and Systems Magazine,* vol. 5, no. 3, pp. 19-29, third quarter 2005.

[8] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *ASPLOSIX: Proceedings of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems*, New York, NY, USA: ACM Press, 2000, pp. 93–104.

[9] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," *SIGOPS Operating System Review*, vol. 34, no. 5, pp. 93–104, 2000.

[10] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, no. 4, pp. 393-422, 2002.

[11] G. Hoblos, M. Staroswiecki, and A. Aitouche, "Optimal design of fault tolerant sensor networks," in *Proceedings of IEEE International Conference on Control Applications*, Anchorage, AK, September 2000, pp. 467–472.

[12] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann, "Scalable coordination for wireless sensor networks: Self-configuring localization systems," in *Proceedings of the 6th International*

Symposium on Communication Theory and Applications (ISCTA 2001), Ambleside, UK, July 2001, pp. 1-6.

[13] C. Chong and S. Kumar, “Sensor networks: Evolution, opportunities, and challenges,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247-56, August 2003.

[14] D. Chen and P. K. Varshney, “QoS support in wireless sensor networks: A survey,” in *Proceedings of International Conference on Wireless Networks (ICWN ’04)*, pp. 227–33, Las Vegas, Nev, USA, June 2004.

[15] Z.J. Li and G. Gong, “A survey on security in wireless sensor networks,” Technical Report, University of Waterloo, CACR 2008-20, October 2008.

[16] S. A. Camtepe and B. Yener, “Key distribution mechanisms for wireless sensor networks: A survey,” Technical Report, Rensselaer Polytechnic Institute, TR-05-07, March 2005.

[17] Y. Wang, G. Attebury, and B. Ramamurthy, “A survey of security issues in wireless sensor networks,” *IEEE Communications Surveys & Tutorials*, vol. 8, no. 2, pp. 2-23, 2006.

[18] Y.C.Wang and Y.C. Tseng, ”Attacks and defenses of routing mechanisms in ad hoc and sensor networks,” *Security in Sensor Networks*, Auerbach Publications, pp. 3-25, 2007.

[19] E. Sabbah and K-D Kang, “Security in wireless sensor networks,” in *Guide to Wireless Sensor Networks*, London: Springer-Verlag, 2009, 491-512.

[20] A.D. Wood and J.A Stankovic, “Denial of service in sensor networks,” IEEE Computer, vol. 35, no. 10, pp. 54-62, 2002.

[21] B. Culpepper and H. Tseng, “Sinkhole intrusion indicators in DSR MANETs,” in *Proceedings of First International Conference on Broadband Networks*, 2004, pp. 681-88.

[22] J. R. Douceur, “The Sybil attack,” in *Proceedings of 1st International Workshop on Peer-to-Peer Systems (IPTPS ’02)*, March 2002, pp. 251-260.

[23] Y. Hu, A. Perring, and D. Johnson, “Packet lashes: A defence against wormhole attacks in wireless ad hoc networks,” in *IEEE Annual Conference on Computer Communications (INFOCOM)*, 2003, pp. 1976–86.

[24] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

[25] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary. “Chapter 17: Wireless sensor network security: A survey,” in *Security in Distributed, Grid, Mobile, and Pervasive Computing*, Auerbach Publications, 2007.

[26] Y.W. Law and M. Palaniswami, “Chapter 20: Key management in wireless sensor networks,” in *Guide to Wireless Sensor Networks*, London: Springer-Verlag, 2009, 513-31.

[27] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” in *9th ACM conference on Computer and Communications Security*, 2002, pp. 41-47.

[28] H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for sensor networks,” in *IEEE Symposium on Research in Security and Privacy*, 2003, pp. 197-213.

[29] J. Hwang and Y. Kim, “Revisiting random key pre-distribution for sensor networks,” in *ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 04)*, 2004, pp. 43-52.

[30] D. Hwang, B. Lai, and I. Verbauwhede, “Energy-memory-security tradeoffs in distributed sensor networks,” in *3rd International Conference on Ad-Hoc Networks and Wireless (ADHOC-NOW 2004)*, 2004, pp. 70-81.

[31] S. Zhu, S. Xu, S. Setia, and S. Jajodia, “Establishing pairwise keys for secure communication in ad hoc networks: A probabilistic approach,” in *11th IEEE International Conference on Network Protocols (ICNP’03)*, 2003, pp. 326-335.

[32] A. Shamir, “How to share a secret,” *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[33] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, “A key management scheme for wireless sensor networks using deployment knowledge,” in *IEEE Infocom’04*, 2004, pp. 586-597.

[34] R. Blom, “An optimal class of symmetric key generation systems,” in *Proceedings of EUROCRYPT 84 - A Workshop on the Theory and Application of Cryptographic Techniques*, Paris, France, April 1984, pp. 335-338.

[35] W. Du, J. Deng, Y. Han, and P. Varshney, “A pairwise key pre-distribution scheme for wireless sensor networks,” in *Proceedings of the 10th ACM conference on Computer and Communications Security CCS’03*, 2003, pp. 42-51.

[36] C. Blundo, A. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, “Perfectly-secure key distribution for dynamic conferences,” in *Advance of Cryptology* (*Crypto 92)*, 1992, pp. 471-486.

[37] D. Liu and P. Ning, “Establishing pairwise keys in distributed sensor networks,” in *10th ACM Conference on Computer and communications security (CCS’03)*, 2003, pp. 41-77.

[38] G. Gaubatz, J. Kaps, and B. Sunar, “Public keys cryptography in sensor networks – revisited,” in *the Proceedings of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS)*, 2004, pp. 2-18.

[39] Q. Huang, J. Cukier, H. Kobayashi, B. Liu, and J. Zhang, “Fast authenticated key establishment protocols for self-organizing sensor networks,” in *the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, 2003, pp. 141-150.

[40] D. Malan, M. Welsh, and M. D. Smith, “A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography,” in *the First Annual IEEE Conference on Sensor and Ad Hoc Communications and Networks*, October 2004, pp. 71-80.

[41] G. Gaubatz, J-P. Kaps, E. Öztürk, and B. Sunar, “State of the art in ultra-low power public key cryptography for wireless sensor networks,” in *the 3rd IEEE International Conference on Pervasive Computing and Communication Workshops (PerComc 2005)*, Hawaii, USA, March 2005, pp. 146-150.

[42] L. Batina, N. Mentens, K. Sakiyama, B. Preneel, and I. Verbauwhede, “Low-cost elliptic curve cryptography for wireless sensor networks,” in *Proceedings of the 3rd European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS 2006)*, Hamburg, Germany, September 2006, pp. 6-17.

[43] D. Liu and P. Ning, “Location-based pairwise key establishment for static sensor networks,” in *the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, 2003, pp. 72-82.

[44] E. Blaß and M. Zitterbart, “Towards acceptable public-key encryption in sensor networks,” in *the ACM Second International Workshop on Ubiquitous Computing (ACM SIGMIS)*, 2005, pp. 1-6.

[45] Q. Jing, J. Hu, and Z. Chen, “C4W: An energy efficient public key cryptosystem for large-scale wireless sensor networks,” in *the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, Oct. 2006, pp. 827-832.

[46] L. Oliveira, M. Scott, J. Lopez and R. Dahab, “TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks,” in *the 5th International Conference on Networked Sensing Systems*, 2008, pp. 173-180.

[47] Q. Dong and D. Liu, “Using auxiliary sensors for pairwise key establishment in WSN,” in *NETWORKING ’07: Ad Hoc and Sensor Networks, Wireless Networks, and Next Generation Internet*, Berlin, 2007, pp. 251-262.

[48] D. Huang, M. Mehta, A. V. Liefvoort, and D. Medhi, “Modeling pairwise key establishment of random key predistribution in large-scale sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1204–1215, Oct. 2007.

[49] R. Anderson and M. Kuhn, “Tamper resistance - a cautionary note,” in *Proceedings of the Second USENIX Workshop on Electronic Commerce*, November 1996, pp. 1–11.

[50] L. Xie, Y. Xu, X. Li; and Y. Zhu, “A lightweight scheme for trust relationship establishment in ubiquitous sensor networks,” in *the Sixth IEEE International Conference on Computer and Information Technology (CIT ‘05)*, 2006, pp. 229-35.

[51] D.W. Carman, P.S. Kruus and B.J.Matt, “Constraints and approaches for distributed sensor network security,” NAI Labs, Technical Report No. 00010, September 2002.

[52] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Chang, “Energy analysis for public-key cryptography for wireless sensor networks,” in *the Proceedings of third IEEE International Conference on Pervasive Computing and Communications,* Italy, March 2005, pp. 324-328.

[53] A. Francillon and C. Castelluccia, “TinyRNG: A cryptographic random number generator for wireless sensors network nodes,” in *the 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2007, pp. 1-7.

[54] Z. Yu and Y. Guan, “A key pre-distribution scheme using deployment knowledge for wireless sensor networks,” in *the 4th Symposium on Information Processing in Sensor Networks,* 2005, pp. 261-268.