Group-based Secure Communication for Wireless Sensor Networks

By

Kashif Kifayat

BSc, Peshawar University, 2003
MSc, University of Liverpool, 2005

Thesis Submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy in Computer Science

School of Computing and Mathematical Sciences

September 2008

Acknowledgements

I would like to take the opportunity to pay thanks to the people who guided and supported me throughout my PhD. Without their support and contribution, successful completion of my research would not have been possible.

I would like to express my gratitude to my first supervisor, Professor Madjid Merabti for his continuous support, encouragement, guidance and invaluable suggestions during this research. I would also like to say special thanks to my second supervisor Dr Qi Shi for, his numerous helpful discussions and his valuable time to help me with various stages of my project, in particular, giving valuable suggestions and comments on my work. Additionally I am deeply indebted to my third supervisor Dr. David Llewellyn-Jones for his sincere guidance and encouragement throughout my research, which has been invaluable and unsparing. Therefore I do want to show my deep appreciation to him.

My thanks also go to the researchers, administration staff and technicians in the School of Computing and Mathematical Sciences at Liverpool John Moores University for their support over the past years.

I would like to show my great appreciation to my beloved parents, brother, sisters and wife for their support and guidance during all these years.

Last but not least, Philip Tarleton, Director of Meade-King, Robinson & Co. Ltd,. also contributed much of his time and efforts to support me during my study. Without him, the work would not have been possible. Thus, I also want to thank him for his support and encouragement.

I dedicate this thesis to my beloved daughter Heba Kifayat.

Abstract

Wireless Sensor Networks (WSNs) are a newly developed networking technology consisting of multifunctional sensor nodes that are small in size and communicate over short distances. Continuous growth in the use of Wireless Sensor Networks (WSNs) in sensitive applications such as military or hostile environments and also generally has resulted in a requirement for effective security mechanisms in the system design. In order to protect the sensitive data and the sensor readings, shared keys should be used to encrypt the exchanged messages between communicating nodes. Many key management schemes have been developed recently and a serious threat highlighted in all of these schemes is that of node capture attacks, where an adversary gains full control over a sensor node through direct physical access. This can lead an adversary to compromise the communication of an entire WSN. Additionally ignoring security issues related to data aggregation can also bring large damage to WSNs. Furthermore, in case an aggregator node, group leader or cluster head node fails there should be a secure and efficient way of electing or selecting a new aggregator or group leader node in order to avoid adversary node to be selected as a new group leader. A key management protocol for mobile sensor nodes is needed to enable them to securely communicate and authenticate with the rest of the WSN.

This thesis presents a new key management protocol stack – entitled the *Structure and Density Independent Group-Based Key Management* protocol (SADI-GKM) – to fulfil the pre-deployment security needs of WSNs. This protocol stack combines four different novel layers with different algorithms. All these layers are integrated with each other to provide better secure solutions for multiple security issues. The first layer of this protocol stack provides structure and density independent key management for large scale WSNs. The second layer's responsibility is to provide secure data aggregation according to the need of the target application. The third layer provides facilities for the secure selection of a new aggregator or group leader sensor node. The fourth layer's main task is to provide key management services for mobile sensor nodes in a WSN. All the four layers of SADI-GKM have been evaluated and implemented using different topologies both with and without group structures and compared against existing solutions. Evaluation results show a significant improvement in terms of resilience against node capture attacks, replication attacks, data confidentiality, secure group leader selection, authentication of mobile sensor node and memory overhead. This shows that our protocol can be used to improve security in WSNs.

List of Abbreviations

ADC	Analogue Digital Converter
ALERT	Automated Local Evaluation in Real-Time
AM	Active Messaging
AP	Asymmetric Pre-distribution
BROSK	BROadcast Session Key
CA	Certificate Authority
DEMA	Differential Electromagnetic Analysis
DGKE	Dynamic Group-Based Key Establishment
DoS	Denial of service
DPA	Differential Power Analysis
DV	Directional Value
EBS	Exclusion Basic Systems
ECC	Elliptic Curve Cryptography
EM	Electro Magnetic
FTSP	Flooding Time Synchronization Protocol
HSN	Heterogeneous Sensor Network
IDS	Intrusion Detection System
ISM	Industrial, Scientific and Medical
LAN	Local Area Network
MAC	Message Authentication Codes
MAN	Metropolitan Area Network
MCU	Microcontroller
MSN	Mobile Sensor Network
NCA	Node Capture Attack
PDFs	Probability Density Functions
RAM	Random Access Memory
RBS	Reference Broadcast Synchronization
RF	Radio Frequency
ROM	Read Only Memory
SADI-GKM	Structure And Density Independent Group-Based Key Management
SEMA	Simple Electromagnetic Analysis
SIA	Secure Information Aggregation
SPA	Simple Power Analysis
SPOT	Small Programmable Object Technology
TPSN	Timing-sync Protocol for Sensor Networks
UTOS	Un-trusted Extension for TinyOS
WAN	Wide Area Network
WBAN	Wearable Wireless Body Area Network
WSNs	Wireless Sensor Networks

Table of Contents

List of Figures

List of Tables

Chapter One: Introduction

With the continuous growth and development of computer networks, the idea of Wireless Sensor Networks (WSNs) introduced by Mark Weiser [1] have received increasing attention. However, research in WSNs is currently in an early stage, so they face many barriers. An important issue is that WSNs must be secure in order to counter a number of security threats from malicious entities. Therefore secure communication is essential. One solution for this is to use a key management scheme to provide secure communication. This is the topic for this thesis.

This chapter is organized as follows. First, the topic of the thesis is presented with its aims. Second, the novel contribution of the new approach posited in the thesis is presented. Third, an overview of the chapters of the thesis is presented. Finally, the chapter is summarized.

1.1 Wireless Sensor Networks

Humans always invent new technologies according to their needs to bring more flexibility to their daily lives. The history of networking is a good example of how computer networking has become more efficient and flexible day by day, e.g. the evolution from wired networks to wireless networks to bring more amenities and flexibility to users.

Correspondingly, WSNs are a newly developed networking technology consisting of multifunctional sensor nodes that are small in size and communicate over short distances. WSNs provide more and unique facilities to users, many of which would be impossible otherwise. Sensor nodes incorporate properties for sensing environments, data processing and communication with other sensors.

WSNs are a new technology that provides facilities for users to monitor environments across a wide area using their laptops or PCs. The unique properties of WSNs increase

flexibility and reduce user involvement in operational tasks such as in battlefields. The role of WSNs in many applications can be very useful, but on the other hand there remain significant challenges for researchers in providing efficient communication and correct information from such networks using reduced resources. In the future WSNs will be an integral part of our lives [2]. They may be working in busy road intersections [7], in the interior of large machineries, at the bottom of an ocean, on the surface of an ocean during a tornado, in a biologically or chemically contaminated field, in a battlefield beyond enemy lines, in a home or large building, in a large warehouse, attached to animals, attached to fast moving vehicles, in a drain or river moving with current.

WSNs are a type of wireless ad hoc network in which communication links are wireless and refer to network connections established for a single session. Such a network does not require a router because every node in the network provides a routing service for others. Minimal configuration and quick deployment make ad hoc networks particularly suitable for emergency situations.

If we compare the basic functionalities of a sensor node with a computer, we find them to be similar. As a sensor node receives input data through sensing, it processes it and produces an output to send on to its destination. Similarly computers receive input from a user, process it and produce output. Consequently we can call them tiny computers with additional sensing capabilities.

1.2 WSN Communication Architecture

A WSN is composed of a large number of sensor nodes and a base station. A base station is typically a gateway to another network, a powerful data processing or storage centre, or an access point for human interfaces. It can be used as a connection to disseminate control information into the network or extract data from it. A base station is also referred to as a sink [116]. Sinks are often many orders of magnitude more powerful than sensor nodes. The sensor nodes are usually scattered in a sensor field and each of these scattered sensor nodes has the capabilities to collect data and route data back to a sink and end users as shown in Figure 1-1. The sink may communicate with the task manager node via

the Internet or via Satellite communications [7]. In a WSN every sensor node plays a role as a router.

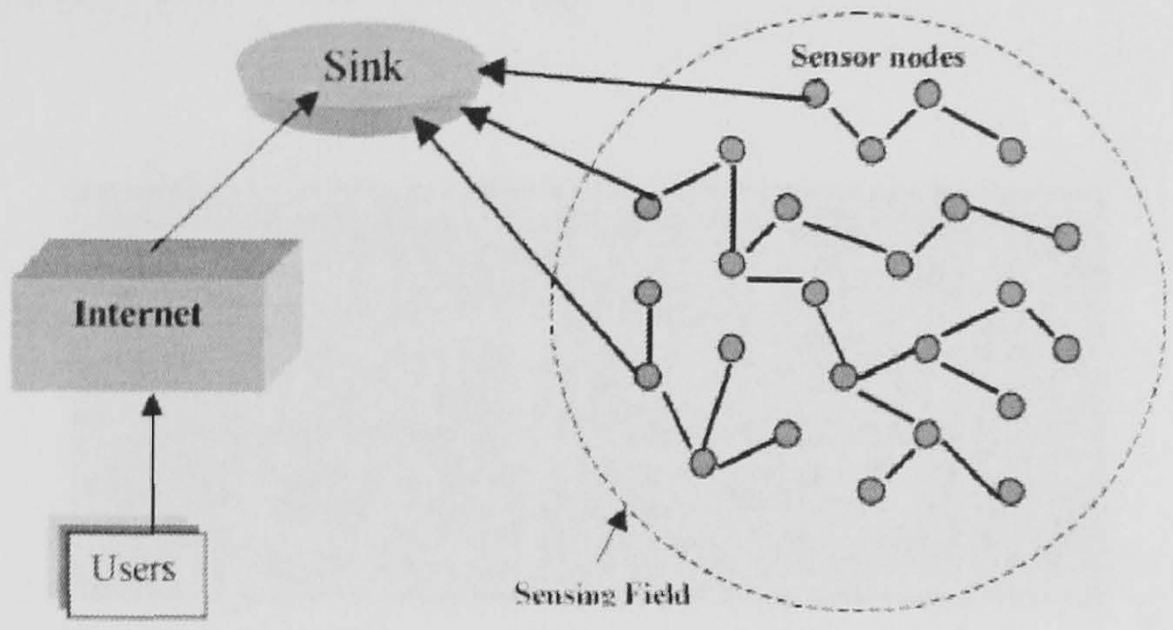


Figure 1-1: Wireless Sensor Network.

WSNs might consist of different types of sensor node such as low sampling rate magnetic, thermal, visual, infrared, acoustic or radar sensors, which are able to monitor a wide variety of ambient conditions [23].

Sensor nodes are densely deployed either very closely or directly inside the phenomenon to be observed. Therefore, they usually work unattended in remote geographic areas. WSNs have different communication patterns for different applications according to their requirements. The categories of these patterns include:

- Node to base station communication, e.g. sensor readings and specific alerts.
- Base station to node communication, e.g. specific requests and key updates.
- Base station to all nodes, e.g. routing beacons, queries or reprogramming of the entire network.
- Communication amongst a defined cluster of nodes (say, a node and all its neighbors) [116]. Clusters can reduce the total number of messages using data aggregation.

1.2.1 Components of a Sensor node

Sensor nodes are capable of gathering sensory information, performing processing and communicating with other connected nodes in the network [179]. The typical hardware architecture of a sensor node is shown in Figure 1-2.

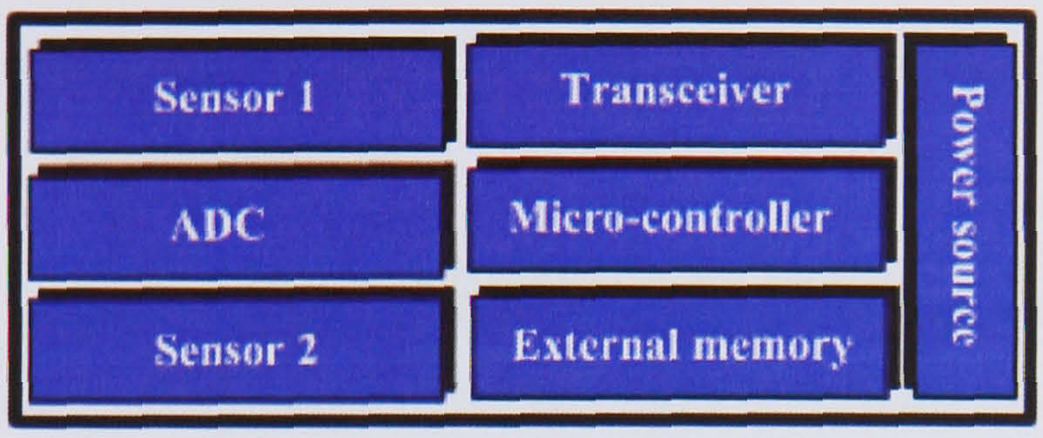
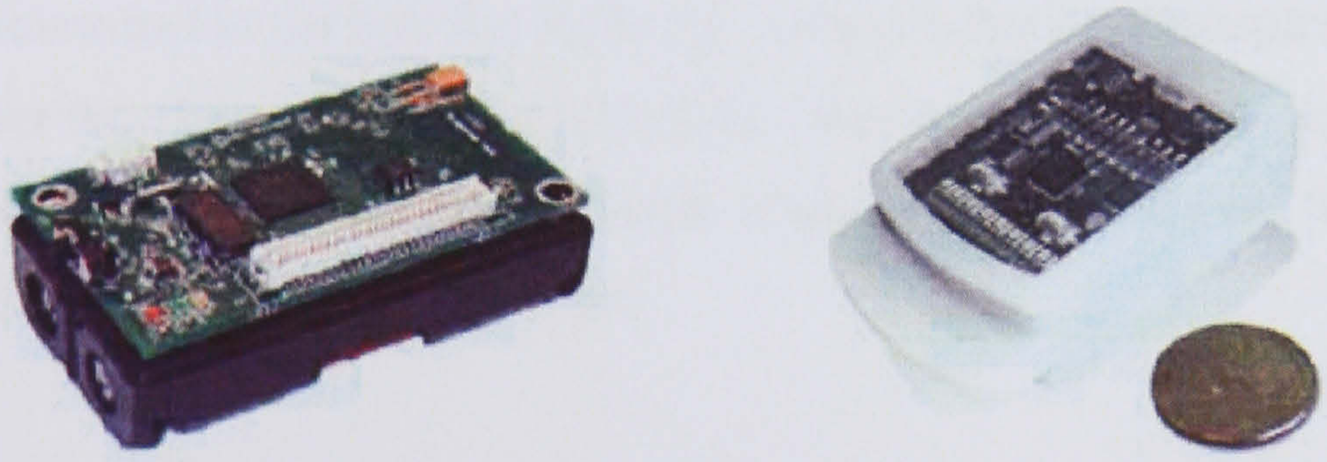


Figure 1-2: Sensor node architecture.

Figure 1-3 (a) shows a crossbow sensor node called mote and Figure 1-3 (b) shows Sun Microsystems' new sensor node called a Sun SPOT.



(a) : Crossbow mote **(b): Sun SPOT**

Figure 1-3: Sensor node devices.

The main components of a sensor node are a Microcontroller, Transceiver, External memory, Analogue to Digital Converter (ADC) and power source. We explain the functionality of the most important elements below.

(a) Microcontroller: A microcontroller (or MCU) is a computer-on-a-chip. It is a type of microprocessor emphasizing self-sufficiency and cost-effectiveness, in contrast to a general-purpose microprocessor (such as the kind used in a PC). The only difference between a microcontroller and a microprocessor is that a microprocessor has three parts – an ALU, a Control Unit and registers (like memory), while the microcontroller has additional elements such as ROM and RAM [179]. The microcontroller performs tasks, processes data and controls the functionality of other components in the sensor node.

(b) Transceiver: A transceiver unit connects the node to the network via wireless radio communication. Sensor nodes make use of the ISM (industrial, scientific and medical) radio band which gives free radio, huge spectrum allocation and global availability [4]. Radio Frequency (RF) based communication is compatible with the majority of WSN applications. WSNs use the communication frequencies between about 433 MHz and 2.4 GHz. Radios used in transceivers operate in four different modes: Transmit, Receive, Idle, and Sleep. Radios operating in the Idle mode result in power consumption almost equal to the power consumed in the Receive mode [5]. Thus it is better to completely shutdown the radios rather than leave them in the Idle mode when not transmitting and receiving. A significant amount of power is also consumed when switching from the Sleep mode to the Transmit mode to transmit a data packet.

(d) Power Source: Sensor node power is consumed by sensing, communication and data processing actions. Communications between sensor nodes consume higher energy as compared to sensing and processing. Sending one bit requires the same amount of energy as executing 50 to 150 instructions on a sensor node [6]. Batteries are the main source of power supply for sensor nodes. Two types of batteries can be used, namely chargeable and non-rechargeable. They are also classified according to the electrochemical material used for the electrodes such as NiCd (nickel-cadmium), NiZn (nickel-zinc), Nimh (nickel metal hydride) or Lithium-Ion [179].

(c) External Memory: A sensor node may make use of memory for two purposes: storing application related or personal data and programming the device. Flash memory is used in sensor nodes due to its cost and storage capacity. The memory size of mica-2 nodes is 128K Flash and 4K Ram.

1.3 Applications of WSNs

WSNs are different from traditional networks and present a new set of properties. Typically the communication structure of a traditional network will remain the same in all its applications while a WSN's structure will change according to its application. WNSs can be classified into two categories according to applications. The first category is that of indoor WSNs and the second is that of outdoor WSNs. Indoor WSNs can be implemented in buildings, houses, hospitals, factories etc [17, 21-23]. Outdoor WSNs can be implemented for battlefield, marine, soil, and atmospheric monitoring; forest fire detection; meteorological or geophysical research; flood detection; bio-complexity mapping of environments; pollution studies; etc [6, 7, 11, 12] [118]. Other applications of sensor nodes can be found in smart environments, interactive museums [21], car theft monitoring [22], inventory control, vehicle tracking and detection [23], soil moisture monitoring, pH and salinity level measurement, traffic control and road detection, aircraft and space vehicles to report excessive temperatures, tire temperature and pressure monitors on automobiles, aircraft to provide early warnings of impending tread separation [7], and many others. We now present some WSN projects for different applications, including:

1. ***PODS-A Remote Ecological Micro-Sensor Network:*** PODS is a research project conducted at the University of Hawaii, which involved building a wireless network of environmental sensors to investigate why endangered species of plants grow in one area but not in neighbouring areas [13].
2. ***Flood detection:*** ALERT (Automated Local Evaluation in Real-Time) was probably the first well-known WSN deployed in the real world. It was developed by the National Weather Service in the 1970's. ALERT provides important real-time rainfall and water level information to evaluate the possibility of potential

flooding. Currently ALERT is deployed across most of the western United States. It is heavily used for flooding alarming in California and Arizona [14].

3. ***ZebraNet:*** ZebraNet is studying power-aware, position-aware computing/communication systems. On the biology side, the goal is to use the systems to perform novel studies of animal migrations and inter-species interactions [95, 96].
4. ***Monitoring Volcanic Eruptions with a WSN:*** Two WSNs on active volcanoes were deployed by this project [90, 91]. Their initial deployment at Tungurahua volcano, Ecuador, in July 2004 served as a proof-of-concept and consisted of a small array of wireless nodes capturing continuous infrasound data. Their second deployment at Reventador volcano, Ecuador, in July/August 2005 consisted of 16 nodes deployed over a 3 km aperture on the upper flanks of the volcano to measure both seismic and infrasonic signals with a high resolution (24 bits per channel at 100 Hz) [89].

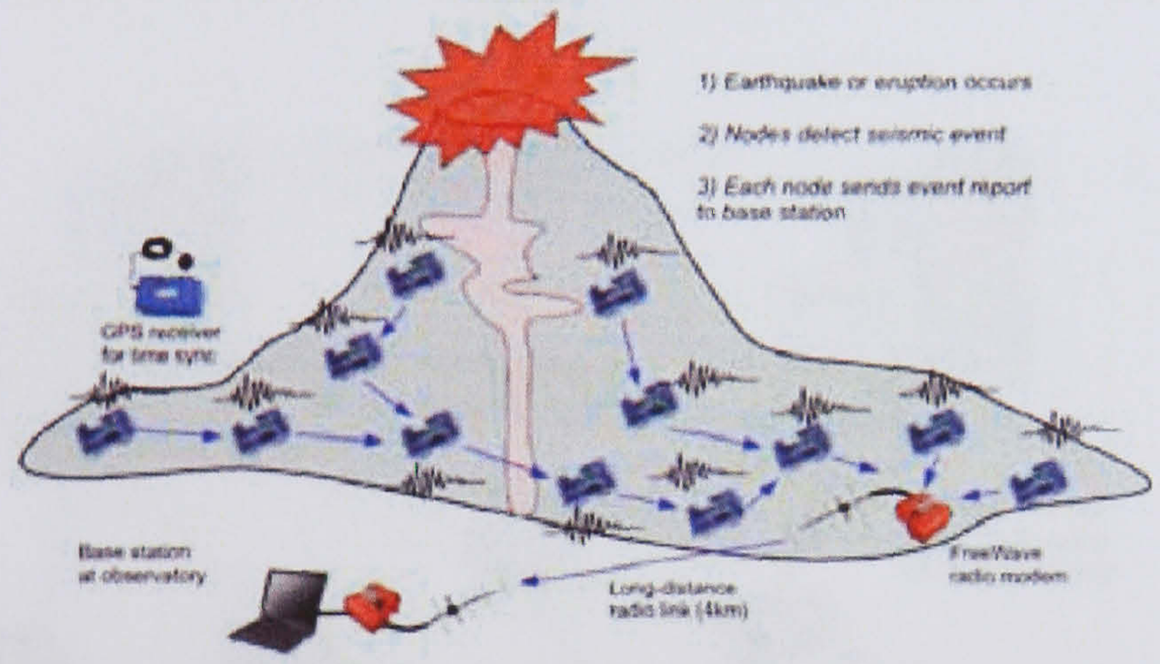


Figure 1-4: Volcano monitoring sensor network architecture [89].

5. ***FireWxNet:*** FireWxNet is a multi-tiered portable wireless system for monitoring weather conditions in rugged wild land fire environments. FireWxNet provides the fire fighting community with the ability to safely and easily measure and view fire and weather conditions over a wide range of locations and elevations within

forest fires [108]. FireWxNet was deployed in summer 2005 at Montana in Colorado.

Figure 1-5: System overview: Radios with directional antennas were used at each peak and at the base camp to relay data from the sensor network and webcam [108].

6. ***Code Blue*:** The Code Blue project research is to apply WSN technology to a range of medical applications, including pre-hospital and in-hospital emergency care, disaster response and stroke patient rehabilitation [15-19, 20]. The Code Blue software platform is shown in Figures 1-6 and 1-7.

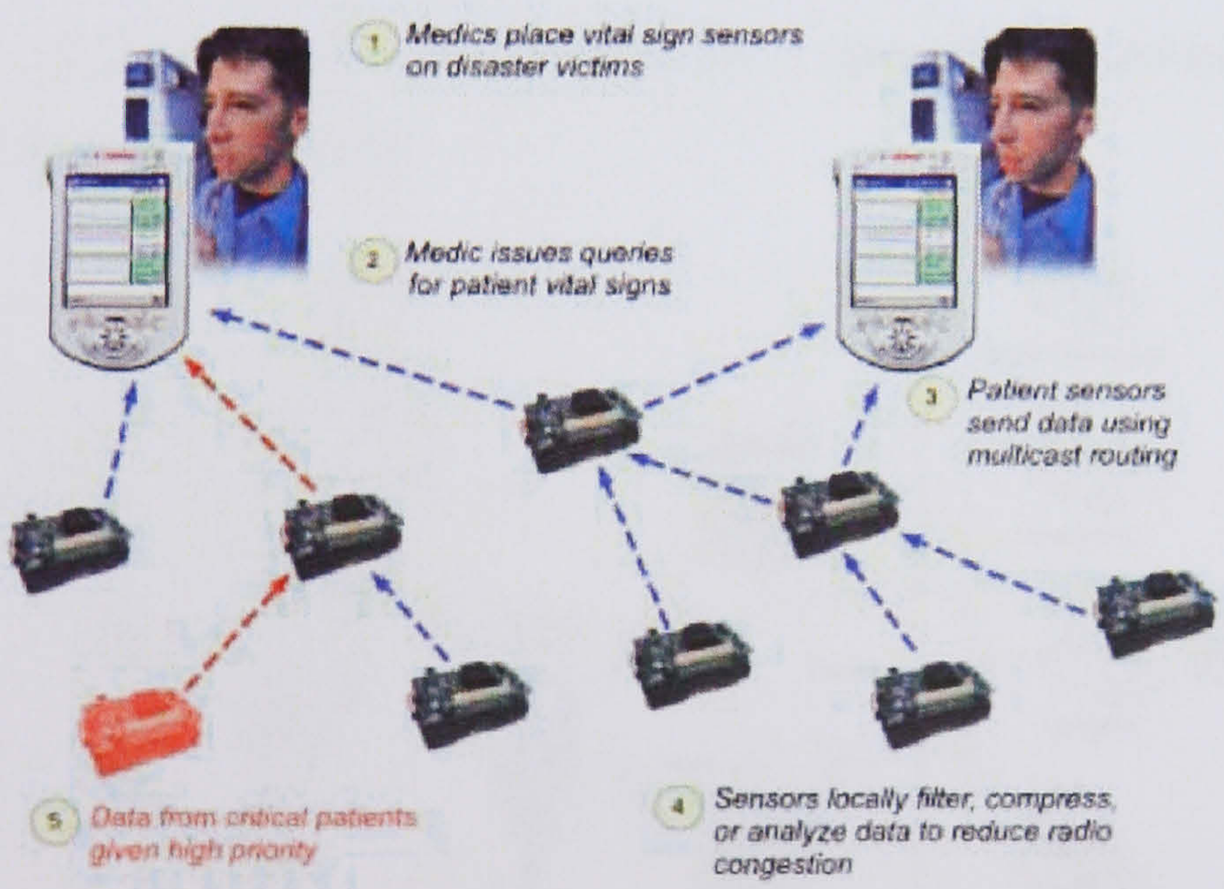


Figure 1-6: Code Blue architecture for emergency response [18].

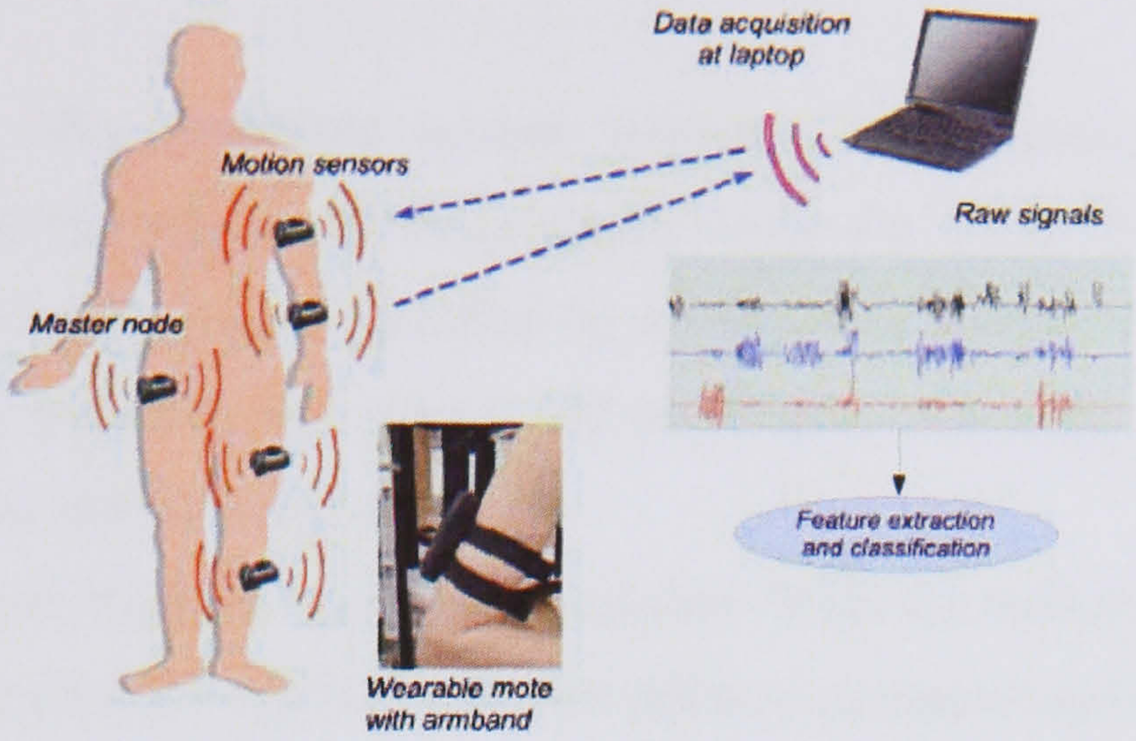


Figure 1-7: Monitoring limb movement in stroke patient rehabilitation [20].

7. ***WBAN*** (Wearable Wireless Body Area Network): The WBAN [24] implementation consists of inexpensive, lightweight, and miniature sensors that can allow long-term, unobtrusive, ambulatory health monitoring with instantaneous feedback to the user about the current health status and real-time or near real-time updates of the user's medical records. Such a system can be used where intelligent heart monitors can warn users about impeding medical conditions [25] or provide information for a specialized service in the case of catastrophic events [19][26].

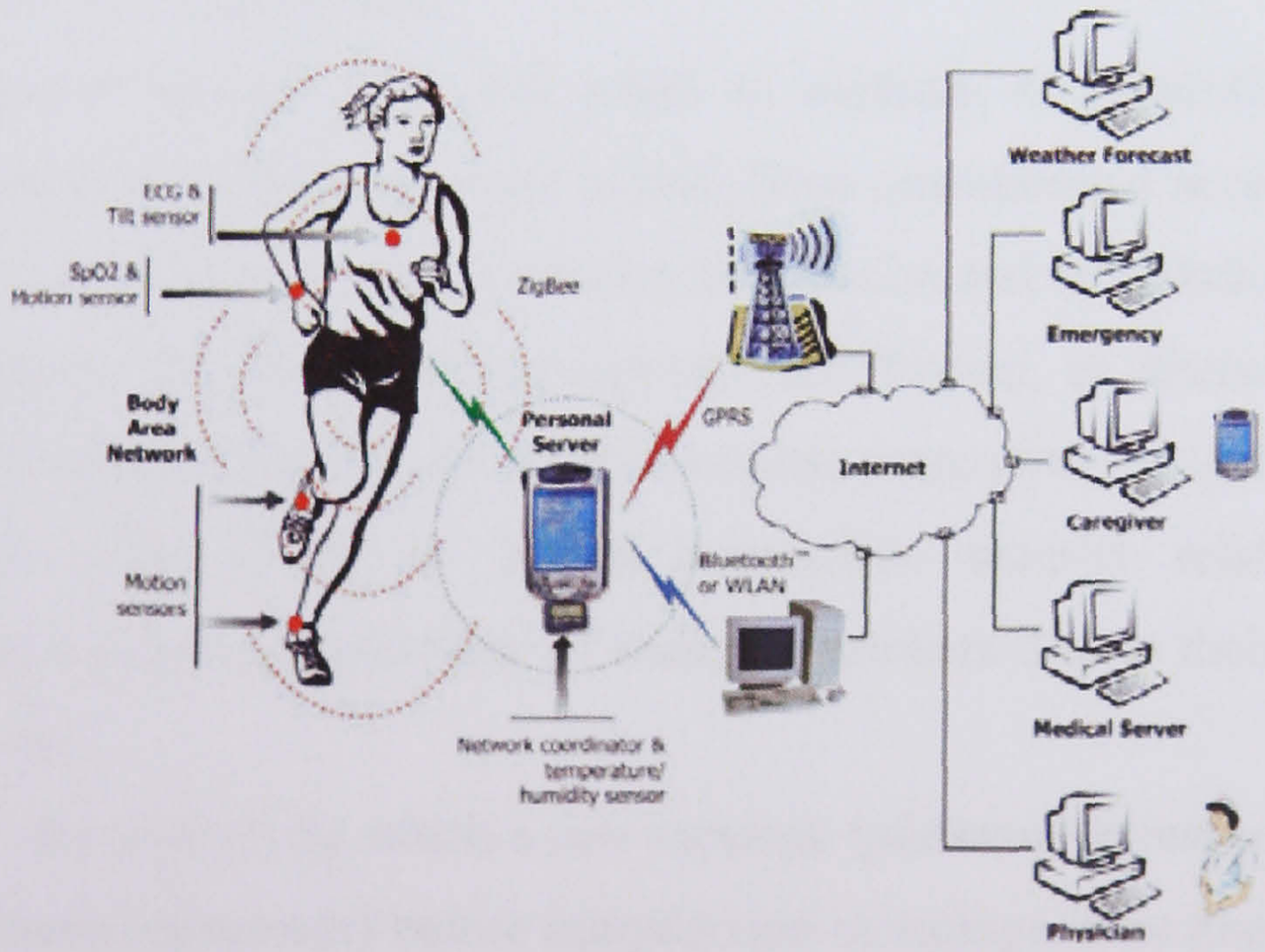


Figure 1-8: Wireless body area network of intelligent sensors for patient monitoring. [24].

8. ***AlarmNet:*** The AlarmNet system integrates heterogeneous devices: some wearable on the patient and some placed inside the living space. Together they perform a health-mission specified by a healthcare provider. Data is collected, aggregated, pre-processed, stored, and acted upon, according to a set of system requirements identified [92-94].
9. ***VigilNet:*** The VigilNet system is a real-time WSN for military surveillance. The general objective of VigilNet is to alert military command and control units of the occurrence of events of interest in hostile regions [8-10].

The above projects showed us the important involvement of WSNs in our daily life. It will not only help us in emergency services to save lives but also benefit us to monitor forest fire, flood detection and many other sectors. We believe in future, we will see WSNs in many new applications to play an integral part of our life.

1.4 Security in WSNs

The main focus of the work presented here is on the development of security mechanisms for WSNs. Before considering such issues in greater detail, it will be useful to first consider security in a wider context.

The term *computer security* generally refers to methods of protecting information, computer programs, and other computer system from unauthorized access, whereas the term *information security* applies to protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction. The core principles of information security are confidentiality, possession, integrity, authenticity, availability and utility [174]. In WSNs information security related issues and requirements have gained the attention of many researchers due to their importance in many applications.

Cryptography – the process by which a raw message (*plaintext*) is mapped or *encrypted* to a scrambled form (*ciphertext*) before transmission or storage, then mapped back to its original form again (*decrypted*) when an authorized party wishes to read it – this is the de facto mechanism used for information security [174]. Encryption and decryption

generally require the use of some secret information, referred to as a *key*. In some encryption mechanisms, the same key is used for both encryption and decryption; whereas for other mechanisms different keys are used. Moreover *key management* deals with the secure generation, distribution, and storage of keys. Secure methods of key management are extremely important. Once a key has been generated, it must remain secret to avoid serious security mishaps [115]. Security is always an issue in traditional networks and brings increasing challenges over time. WSNs have similar and additional issues as compared to traditional networks. Below we provide a brief description of some of the security issues and goals in WSNs that are addressed in this thesis [41]:

Data Confidentiality: Data confidentiality is an issue in network security. In WSNs confidentiality relates that it should not leak sensor readings to its neighbours, build a secure channel for secure communication and public sensor information should be encrypted, such as sensor identities and public keys. Generally key establishment is used to achieve data confidentiality. For example Jun et al. [175] has proposed a solution using symmetric key establishment to protect confidentiality against a parasitic adversary. Many current proposed solutions for data confidentiality suffer from a number of problems which we discuss and address in this thesis.

Data Integrity: This is defined as the quality of correctness, completeness, wholeness, soundness and compliance with the intention of the creators of the data. In WSNs secure data aggregation techniques are used to achieve data integrity and data confidentiality [148, 149]. However in applications where sensitive data are collected, some systems have also been proposed to check data integrity using Intrusion Detection Systems [176]. This thesis also focuses on issues related to secure data aggregation, describe these limitations and discusses how they can be mitigated using our novel protocol.

Availability: The accessibility of a system resource in a timely manner; Availability is one of the six fundamental components of information security. The requirement of security not only affects the operation of the network, but also is highly important in maintaining the availability of the whole network. For better availability in WSNs good network management, monitoring and a reliable transport layer solution are needed [177].

Authenticity: Defined as the verification and integrity of a transmitted message. In WSNs an adversary is not just limited to modifying the data packet. It can change the whole

packet stream by injecting additional packets. So the receiver needs to ensure that the data used in any decision-making process originates from the correct source. Many schemes have been proposed and in this thesis we draw on the benefits they provide to provide an authentication scheme to provide increased information security.

These general issues directly affect the functionality of our proposed protocol and are addressed in later subsections.

The limited resources of a sensor node and its different characteristics from those of a traditional computer make it difficult to use traditional security techniques for WSNs.

As described earlier, a sensor node is a tiny device, with only a small amount of memory and storage space for the code. In order to build an effective security mechanism, it is necessary to limit the code size of security algorithms. For example, one common sensor type (TelosB) has a 16-bit, 8 MHz RISC CPU with only 4-10K RAM, 48K program memory, and 1024K flash storage [110]. With such a limitation, the software built for the sensor node must also be quite small. Therefore, the size for all security related code must be small. Security also gets more challenging when we talk about scalable WSNs or add considerations of mobility to the WSNs. During our research we have identified that even topologies directly affect security as well [33]. All these issues are inter-linked with each other, making them even more challenging. In the following subsection we formally define the problem domain.

1.5 Problem Definition

A WSN is vulnerable to several security threats/attacks similar to those of traditional networks. One such possible attack on WSNs is called a *node capture attack* (NCA): where an adversary gains full control over a sensor node (or nodes) through direct physical access. This can lead to compromise of the communication of an entire WSN. Such an attack could allow an adversary to launch many other attacks, e.g. Replay, Blackhole and Denial of Service attacks. The compromised sensor node can be an aggregator node, a cluster head node or a normal sensor node. Therefore data confidentiality and data integrity are at high risk. Many key management solutions have

been proposed recently to provide secure communication and reduce the impact of adversaries damaging WSNs using node capture attacks, but these solutions still suffer from its effects. However there is also the possibility of replay attacks without physically compromising a sensor node, where an adversary node will eavesdrop on packets and resend old packets in an attempt to constantly waste other sensor nodes' energy or obtain secret keys.

In WSN many applications are carrying sensitive data and information and thus we should consider threat like node capture attack as high risk for communication and data confidentiality/security. Furthermore, ignoring security issues relating to data aggregation can result in significant damage of data confidentiality to a WSN. Although data aggregation in WSNs is helpful in reducing the amount of data to be transmitted between sources and destinations – thereby conserving energy, it is important to balance this against the security implications. Current data aggregation schemes are designed without considering possible security issues related to data confidentiality. Therefore data aggregation needs to be enhanced to maintain the privacy/security of sensor nodes and their data. Additionally in case of aggregator node, group leader or cluster head node fails due to fewer resources or physical compromise there should be a secure and efficient way of electing or selecting a new aggregator or group leader node. For example a new selected group leader node can be an adversary node. There are different proposed schemes which are only providing election of a new group leader node but they didn't consider related security issues. Therefore an efficient secure group leader/cluster head/aggregator node selection solution is one of the important requirements of WSNs

The sensor nodes' mobility poses far more challenges in Mobile Sensor Networks (MSNs) compared to Static Sensor Networks. The network topology is highly dynamic as sensor nodes frequently join or leave the WSN, and roam throughout the WSN. The wireless channel is also subject to greater interferences and errors, revealing volatile characteristics in terms of bandwidth and delay. With such dynamics, mobile nodes may request for anytime, anywhere security services as they move from one place to another.

In terms of solutions to these security challenges in Mobile and Static WSNs, researchers have proposed different key management schemes [2-9, 14] for secure communication

and resilience against possible attacks. These schemes try to provide better resilience against node capture attacks, but there is still a chance of the entire WSN being compromised. For example, in probabilistic key pre-distribution schemes [2, 5], compromising a few nodes can lead to the entire WSN communications being compromised.

One of the important concerns in terms of security solutions for WSNs is that all these proposed solutions are specifically designed for single application or security problems, explicitly with certain attack models in mind. They work well in the presence of a designated network model, application or attack but may collapse under unanticipated attacks, different application or changes to the WSN model. Therefore we are unaware of the consequences when multiple security problems occur together in a WSN. To achieve this goal, a new application independent and integrated approach is needed.

A challenge remains therefore to create an application independent, scalable and integrated key management solution for WSNs which can provide better secure communication, secure data aggregation, data confidentiality, and resilience against node capture and replication attacks for different applications. Furthermore such a proposed key management solution should also support applications with mobility. Giving this challenge, our intention is to develop a secure communication solution for large scale WSNs, which will be elaborated in the next section.

1.6 Project Aims and Objectives

The aim of this project has been to design a structure and density independent key management solution for large scale WSNs, which can provide secure communication between sources and destinations, resilience against node capture and replication attacks, secure data aggregation, secure group leader selection, and key management in different applications including for MSNs.

Specifically, the objectives of this project are:

- To establish the background related to security issues and security requirements for WSNs, for which existing key management techniques offer appropriate

solutions. This literature research presents current efforts and contributions towards WSN security, and also magnifies the core problems and difficulties faced by the existing key management solutions. This objective has been fulfilled using an extensive literature survey in Chapters two and three.

- To design an application independent, scalable and integrated group-based key management protocol for large scale WSNs, providing secure communication between source and destination nodes. The protocol should be able to:
    - Provide high resilience against node capture attacks to minimize their impact so that compromised sensor nodes should not help an adversary to compromise the communication among other sensor nodes.
    - Provide resilience against replay attacks to ensure data freshness.
    - Minimize memory usage and communication overhead, and increase key connectivity.

We present the protocol that we have developed to fulfil these objectives in section 5.2 and evaluated in section 6.3

- To provide different levels of data confidentiality. Consequently the proposed protocol should provide a secure data aggregation service at the sensor node acting as a group leader. This service will help to ensure that sensitive data is not disclosed in case a sensor node or group leader node is compromised. The level of data confidentiality should be variable according to application requirements in respect of available resources. We have developed a secure data aggregation scheme which is the second layer of our proposed protocol, described in section 5.2.1 and evaluated in section 6.3.3.

- To design a secure and efficient group leader selection algorithm in case a group leader battery life runs out or is physically compromised. This allows a selection process to be activated to establish a new group leader when certain parameters of the current group leader fall below a given threshold or it is detected to be compromised. We have achieved this by testing various selection parameters in section 5.1.2 that we used to propos an efficient group leader selection scheme

described in section 5.2.2. Finally the proposed scheme is evaluated in section 6.3.8.

- To propose a key management algorithm able to provide a solution for applications where sensor nodes are mobile. We have achieved this by extended our protocol in section 5.2.3 and evaluated in section 6.3.9.

1.7 Novel Research Contributions

This thesis proposed a novel and distinct protocol stack Structure And Density Independent Group Based Key Management (SADI-GKM) for WSNs. This protocol comprises four layers, which we refer to as the protocol stack. Each layer contains its own set of algorithms to perform its own set of functions. Therefore the stack and its algorithms form the first layer provides a structure and density independent key management solution for large scale, the second deals with secure data aggregation, the third operates secure group leader selection services, and the fourth layer offers a key management solution for MSNs. Specifically our novel contributions can be summarized as follows:

- **Development of the structure and density independent key management protocol:** Our proposed key management protocol is designed for large-scale WSNs. As this protocol is topology independent, it can work on different topologies. The protocol has been evaluated using different topologies both with and without group structures and compared against existing key management schemes. Our evaluation results show a significant improvement in terms of resilience against node capture attacks, data confidentiality, memory overhead and connectivity. There are a verity of key management protocols that are available but we have demonstrated using simulation that they are not structure and density independent in section 6.3.1. These protocols are limited to specific applications and a structure and density independent key management solution therefore provides benefits in terms of universal applicability and helps to reduce computation costs in maintaining the network topology when new nodes join or

leave a group/network. Structure And Density Independent Group-Based Key Management (SADI-GKM) as presented in Chapter 5, is therefore a novel protocol.

- **Design of a secure data aggregation algorithm:** This provides secure data aggregation by using homomorphic encryption to aggregate encrypted data without the need for decryption at the group leader node. The algorithm helps to maintain better data confidentiality. Furthermore we have provided two different aggregation solutions according to the level of data confidently/security required in a target application. Current secure data aggregation schemes do not provide different levels of confidentiality. However our secure data aggregation solution provides different levels of confidentiality using different encryption methods for different applications according to their requirements and available resources, therefore providing novel functionality as presented in section 5.2.1.

- **Development of a secure group leader selection algorithm:** This forms part of our key management protocol. This novel method allows a new group leader to be selected using four different weighting factors: available energy at each sensor node, its level of trust, its distance from the current group leader (position of new group leader sensor node) and the number of its neighbouring sensor nodes, presented in section 5.2.2. The sensor node with the highest combined factor value will be selected as the new group leader node. Moreover, the algorithm only involves a few sensor nodes in the new group leader selection process, which helps to reduce the overall cost. Furthermore we have analysed and found the impact of the group leader position on group lifetime in section 5.1.2. Current group leader election/selection schemes are based on simple selection factors such as energy and number of neighbouring sensor nodes. The inclusion of group leader position in our group leader selection protocol is therefore a useful and novel addition.

- **Design of a key management algorithm for MSNs:** There are many different application scenarios for MSNs, where all sensor nodes can be mobile or some of them are static and the others mobile. Furthermore there are different types of

roaming for mobile sensor nodes: free roaming (e.g. WSNs in water) and guided roaming. These properties of MSNs pose more security challenges than static sensor networks. If we link the issue of scalability to MSNs, security issues becomes more complicated. According to our literature survey there is no appropriate key management protocol for WSNs, which can consider all these issues together. Therefore we have proposed a novel key management algorithm for MSNs to provide better secure communication which is presented in section 5.2.3.

- **Discovery of the effect of node topology, density and level of key sharing on security:** During our research we identified that sensor node topology, density (the number of neighbouring nodes) and level of key sharing with neighbouring sensor nodes have a direct effect on WSN security in particular in relation to node capture attacks. Currently the effect of node capture attacks on entire WSNs has not been analysed in the context of topology, density and level of key sharing. We evaluated these factors in section 5.1.1 and show them to have an impact on the security of entire WSN. Therefore we need to be careful in selecting an appropriate topology and density, and consider the number of keys shared between neighbouring nodes [33]. These findings are of significant benefits to the future development of WSN security technologies.

- **Discovery of the effect of group leader position on performance of a sensors group:** During our investigatory research we identified that group leader position in a group has direct effect on the performance of the sensor group. Especially in case of large group of sensor nodes or large packet size the wrong position can increase communication overhead dramatically. Currently group leader position has not been analysed in context of group or WSN lifetime. We evaluated this factor in section 5.1.2 and show group leader position impact on performance of entire group.

1.8 Thesis Structure

Chapter one: In this chapter we discuss the wider context and outline the problem of secure communication in large scale WSNs. We include the definition of WSNs, their main application areas and a survey of current WSN projects. We describe the communication architecture and components of sensor nodes. We also briefly describe the importance of security in WSNs. Furthermore the chapter highlights the consequences of node capture attacks, other interlink attacks and security vulnerabilities.

Chapter two: In this chapter we provide a general overview of challenges in WSNs, including: fault tolerance, sensor network topology, routing, mobility and scalability. Furthermore we present a survey of the security challenges and possible attacks in WSNs. The main security challenges include: data confidentiality, data integrity, authentication, key establishment, availability, privacy, secure routing, secure group management, intrusion detection and secure data aggregation. Possible attacks on WSNs include: node capture attacks, replay attacks, side channel attacks, Denial of Service (DoS) attacks, software attacks, routing attacks, traffic analysis attacks, Sybil attacks and attacks on in-network processing. At the end of chapter we provide an overview of security in MSNs.

Chapter three: This chapter presents a critical survey on literature and works relating to key management, secure data aggregation, group leader election/selection and key management for MSNs. It also includes discussions on the existing solutions of key management for static WSNs. These solutions are classified into five different types: key pool based key management, session based key management, hierarchical based key management, key management for heterogeneous sensor networks and group based key management. Limitations and drawbacks of the current key management schemes are assessed. In particular, all the existing solutions are structure dependent, and any change to the structure of a WSN directly affects its security.

Chapter four: This chapter presents the basic design of our novel Structure And Density Independent Group-Based Key Management (SADI-GKM) protocol stack with a description of its four layers. The chapter starts with the background of current security issues, their relationships and current requirements of secure WSNs. It then highlights the

importance of providing an integrated proactive security solution and explains how our four protocol layers are integrated with each other in order to achieve our goals.

Chapter five: This chapter describes the design and different operation steps of our protocol in details. The important part of this chapter is our pre-design research investigations which help us in developing efficient protocol SADI-GKM. The chapter also describes each layer of our protocol. We begin by looking at the first and second layers, *key management* and *secure data aggregation,* where we describe the basic key management and secure data algorithms together. Then we explain the third layer, *secure group leader selection*. In the fourth layer we present a key management solution for MSNs.

Chapter six: This chapter presents the implementation phases, simulation environments, analysis, results and performance evaluation of our protocol. We start this chapter by describing the implementation phases including a radio model, topology implementations, routing algorithms and security. We evaluate SADI-GKM performance against node capture attacks using various topologies both with and without the use of groups. We then describe our implementation of secure data aggregation using homomorphic encryption, and secure group leader selection. Finally we evaluate key management for MSNs.

Chapter seven: We conclude our dissertation by summarizing the findings that we have achieved so far, and discuss major issues and future work in the area of WSN security.

1.9 Summary

This chapter has presented an overview of this thesis. WSNs consist of a large number of low-cost, low-power, and multifunctional sensor nodes that communicate over short distances through wireless links. Continuous growth in the use of WSNs in sensitive applications such as military or hostile environments has resulted in a requirement for effective security mechanisms in the WSN design. Achieving security in resource-constrained WSNs is a challenging research task. Many key management schemes have been developed recently to provide secure communication between sources and

destinations in WSNs. A serious threat highlighted in all these schemes is that of node capture attacks, where an adversary gains full control over a sensor node through direct physical access. The compromised sensor node can be an aggregator node, a cluster head node or a normal sensor node. This could allow the adversary to compromise the communication of an entire sensor network, which causes a high risk for data privacy. Furthermore ignoring security issues related to data aggregation and aggregator node election can bring large damage to WSNs. To deal with these issues we have therefore proposed the novel protocol SADI-GKM to provide better secure communication, secure data aggregation, privacy, and resilience against node capture and replication attacks.

Chapter Two: Security in Wireless Sensor Networks

WSNs are a new research area that has a large number of complicated research challenges in security, routing, topology management, mobility and many others. In this chapter we will first describe some general research challenges and later concentrate on the security challenges and security threats that exist for WSNs.

2.1 Challenges in WSNs

In this section we will briefly describe general research issues and challenges in WSNs. These include fault tolerance; scalability; production costs; operating environment; topology maintenance; hardware constraints; power consumption; security and so on.

2.1.1 Fault Tolerance

As described in section 1.3, WSNs are often deployed in inhospitable environments. Furthermore sensor nodes need to be inexpensive to achieve target benefits anticipated from their future use. Consequently they are liable to faults and resource depletion.

Fault-tolerance is the ability of a system to deliver a desired level of functionality in the presence of faults. Fault-tolerance is crucial for many systems and is becoming vitally important for computing and communication based systems. Since WSNs are inherently fault-prone and their on-site maintenance is infeasible, scalable self-healing is crucial for enabling the deployment of large-scale WSNs applications [27]. The level of fault tolerance can be higher and lower depending on the different applications of particular WSNs, and relevant schemes must be developed with this in mind. Fault tolerance can be

addressed at the physical layer, hardware, system software, middleware, or application level [28].

2.1.2 WSNs Topology

WSNs might contain a large number of sensor nodes in applications where networking would otherwise be inaccessible. They are also prone to frequent failures and thus make topology maintenance a challenging task. There are generally three main topologies in WSNs. These are grid, tree and random mesh topologies. WSN applications are generally topology dependent. Therefore it can be difficult to make use of any proposed solution related to routing, security and so on across multiple applications. Use of these topologies varies according to applications [33]. The node density may be as high as 20 nodes/m^2, or in some circumstances even higher [32]. Deploying a large number of sensor nodes densely requires careful handling of topology maintenance. Therefore topology maintenance schemes are required. Topology maintenance can be split into several phases: *Pre-deployment, deployment, Post-deployment* and *Re-deployment phases.*

2.1.3 Routing

Routing in WSNs is very challenging due to the inherent characteristics that distinguish these networks from other wireless networks like mobile ad hoc networks or cellular networks. First, due to the relatively large number of sensor nodes, it is not possible to build a global addressing scheme for the deployment of a large number of sensor nodes as the overhead of identification (ID) maintenance is high. Thus, traditional IP-based protocols may not be applied to WSNs [35]. Furthermore, sensor nodes are deployed in an ad hoc manner. According to Al-Karaki and Kamal [35] the design of routing protocols is influenced by many design factors. The following factors must be considered to achieve efficient communication in WSNs: *node deployment, energy consumption without loss of accuracy, data reporting model and node/link heterogeneity.* Fault tolerance, scalability and connectivity are other factors which have a direct influence on routing as described in earlier sections.

2.1.4 Mobility

Mobility is generally viewed as a major hurdle in the control and management of large-scale wireless networks. In fact without mobility (i.e. stationary nodes only), a hierarchical clustering and addressing scheme (of the type used in the Internet) could be easily applied to manage routing. However, as nodes move, the hierarchical partitioning structure also changes, forcing frequent hierarchical address changes followed by update broadcasts to the entire network. This is a very resource consuming proposition that can easily congest the entire network. Most of the network architectures assume that sensor nodes are stationary. However, the mobility of either base stations or sensor nodes is sometimes necessary in many applications [37].

Furthermore mobility in WSNs brings similar (like other wireless networks) and more complicated challenges (related to security, scalability, routing, network management and so on) due to limited available resources and structure dependent applications. Since we are unable to use traditional wireless networks solutions in WSNs due to limited resources and the dense nature of the networks, similar conditions apply to MSNs [36].

2.1.5 Scalability

Generally WSNs are assumed to contain hundreds or thousands of sensor nodes. However the number of sensor nodes depends on applications, and in some circumstances it might reach to millions. Consequently WSNs must be highly scaleable networks and any new scheme must also be able to work in such large-scale WSNs. Scalability is one of the core challenges in WSNs because we need to pay particular attention to the provision of a solution for scaleable routing, security and management when networks are scaleable. Providing these solutions in the limited resource environment of WSNs is a considerable research challenges.

In the management of such large scale networks with high density of neighbouring nodes, every single node plays an important role. This can cause the overloading of individual sensor nodes, which can directly affect the performance of the entire WSNs. The right selection of density can help to balance energy consumption in the network. The density

can range from a few sensor nodes to several hundred sensor nodes in a region [34]. The node density depends on the application in which the sensor nodes are deployed. For example, for machine diagnosis applications, the node density can be around 300 sensor nodes in a 5×5 m^2 region, and the density for vehicle tracking applications can be around 10 sensor nodes per 5×5 m^2 region [32, 7].

2.1.6 Other Issues

There are many other issues such as production costs [23, 30, 31], time synchronization [111, 112], group management [113], boundary recognition [114] and security, along with various other issues related to specific applications. In the next section we will discuss security issues and challenges in WSNs.

2.2 Security Challenges in WSNs

In this section we describe generally and briefly about challenges in WSNs, including data confidentiality, data integrity, authentication, key establishment, availability, privacy, secure routing, secure group management, intrusion detection and secure data aggregation.

2.2.1 Data Confidentiality

In order to secure data from eavesdroppers it is necessary to ensure the confidentiality of sensed data. To achieve data confidentiality, encryption functions are normally used, which are a standard method and rely on a shared secret key existing between communicating parties. To protect the confidentiality of data, encryption itself is not sufficient; as an eavesdropper can perform traffic analysis on the overheard ciphertext, which could release sensitive information about the data. Furthermore, to avoid misuse of information, confidentiality of sensed data also needs to be enforced via access control policies at base stations [40]. To maintain better confidentiality we should follow some of the following rules [38, 39]:

- A WSN should not leak sensor readings to its neighbours. In some applications, the data stored in a sensor node may be highly sensitive. To avoid leakage of sensitive data a sensor node should therefore avoid sharing keys used for encryption and decryption with neighbouring nodes [33].
- Secure channels should be built into WSNs.
- Public sensor information such as sensors' identities should also be encrypted to some extent to protect against traffic analysis attacks.

Physical node compromise makes the problem of confidentiality complex. When an adversary physically captures a sensor node, it is generally assumed that the adversary can extract all information or data from that sensor node. To minimize the risk of disclosing sensitive data after physical attacks on sensor nodes, it is better to use rules described earlier in this section. Further details about node capture attacks will be provided in Section 2.3.1.

In MSNs, higher risk levels are associated with data confidentiality than in static sensor networks due to their roaming and the sharing of information with sensor nodes. Therefore it is particularly important that mobile sensor nodes should not leak sensor readings to neighbouring nodes without proper security. We do not recommend to share keys with neighbouring sensor nodes which are used for data encryption and decryption [36].

2.2.2 Data Integrity

Data integrity issues in wireless networks are similar to those in wired networks. Data integrity ensures that any received data has not been altered or deleted in transit. We should keep in mind that an adversary can launch modification attacks when cryptographic checking mechanisms such as message authentication codes and hashes are not used. For example, a malicious node may add some fragments or alter the data within a packet. This new packet can then be sent to the original receiver [41].

We also need to ensure the freshness of each message. Informally, data freshness suggests that the data is recent, and it ensures that no old messages have been replayed. This requirement is especially important when there are shared-key strategies employed in the design.

2.2.3 Authentication

Authentication is a process which enables a node to verify the origin of a packet and ensure data integrity. In WSNs an adversary is not just limited to modifying data packets. It can change the whole packet stream by injecting additional packets. So the receiver node needs to ensure that the data used in any decision-making process originates from correct sources [41]. In many applications authentication is essential due to matters of sensitivity.

However whilst authentication stops outsiders from inserting or spoofing packets, it does not solve the problem of physically compromised sensor nodes. As a compromised sensor node contains the same secret keys as a legitimate node, it can authenticate itself to the network and an adversary may also exploit the broadcast authentication capabilities of the compromised sensor nodes to attack the WSN itself (e.g. to consume sensors' battery power by instructing them to do unnecessary operations). We may be able to use intrusion detection techniques [40] to spot such compromised nodes, and revoke the broadcast authentication capabilities of the compromised senders [43]. There are many authentications schemes [39, 41, 43-48] that have been proposed for WSNs.

Establishing efficient authentication in MSNs is a more challenging task than in static WSNs. In a static WSN every sensor node might have a fixed number of neighbours, and new sensor nodes are unlikely to be added after deployment. However in a MSN nodes easily roam from one place to another. Providing authentication in large scale MSNs is challenging due to resource limitations [36].

2.2.4 Key Establishment

Key management constitutes a set of techniques and procedures supporting the establishment and maintenance of keying relationships between authorized parties. There are two types of key algorithms. Symmetric key algorithms represent a system involving two transformations: one for a source/sender and another for the receiver, both of which make use of either the same secret key (symmetric key) or two keys easily computed by each other. Asymmetric key algorithms represent a system comprised of two related transformations: one defined by a public key (the public transformation), and another defined by a private key (the private transformation). Finding the private key from the public key can be difficult.

Confidentiality, entity authentication, data origin authentication, data integrity, and digital signatures are some of the cryptographic techniques for which key management performs a very important role.

In indoor and outdoor WSN applications, communications can be monitored and nodes are potentially subject to capture and surreptitious use by an adversary. For this reason cryptographically protected communications are required. A keying relationship can be used to facilitate cryptographic techniques, whereby communicating entities share common data (*keying material*). This data may include public or secret keys, initialization values, or additional non-secret parameters [3].

Many researchers have proposed different key management schemes for secure communication between sensor nodes that try to provide better resilience against node capture attacks, but at some level of node capture attack there is a possibility that the entire sensor network may become compromised. For example, in probabilistic key pre-distribution schemes [2, 5] the compromise of just a few nodes can lead to a compromise in the communications of the entire sensor network.

In general, resource usage, scalability, key connectivity and resilience are conflicting requirements; therefore trade-offs among these requirements must be carefully observed [8]. In the next chapter we will describe various key management techniques and related work in detail.

2.2.5 Availability

Providing availability requires that a sensor network should be functional throughout its lifetime. However, strict limitations and unnecessary overheads weaken the availability of sensors and sensor networks. The following factors have a particular impact on availability [41]:

- Additional computation consumes additional energy. If no more energy exists, the data will no longer be available.
- Additional communication also consumes more energy. What's more, as communication increases so too does the chance of incurring a communication conflict.

Therefore by fulfilling the requirement of security we can help to maintain the availability of the whole network. Denials of service (DoS) attacks such as jamming usually result in a failure of availability. Jamming occurs when a malicious user deliberately derives a signal from a wireless device in order to overwhelm legitimate wireless signals. Jamming may also be inadvertently caused by cordless phones, microwave ovens or other electromagnetic emissions. Jamming results in a breakdown in communications because legitimate wireless signals are unable to communicate on the network [42].

Loss of availability may have serious impacts. In some applications, e.g. manufacturing monitoring applications, loss of availability may cause failures to detect a potential accident resulting in financial loss or even human harm. Loss of availability may also open a back door for enemy invasion in battlefield surveillance applications [40]. Lack of availability may affect the operation of many critical real time applications such as those in the healthcare sector that require a 24 hours operation, the failure of which could even result in loss of lives.

2.2.6 Privacy

The main purpose of privacy in WSNs is to ensure that sensed information stays within the WSNs and is only accessible by trusted parties.

Common approaches generally address concerns of data privacy and location privacy [50-52]. For example, privacy policies govern who can use an individual's data and for which purposes. Furthermore, confidentiality/secrecy mechanisms [53] provide access to data without disclosing private or sensitive information. However, data is difficult to protect once it is stored on a system [49].

An adversary could mount the following attacks to compromise privacy in a network [49]:

- The adversary could simply listen to control and data traffic. Control traffic conveys information about the sensor network configuration. Data traffic contains potentially more detailed information than that accessible through the location server.
- An increase in the number of transmitted packets between certain nodes could signal that a specific sensor has registered its activity.
- A malicious node could trick the system into reducing data distortion (privacy protection) through subject spoofing.
- An inserted or compromised node could drop packets, forward them incorrectly, or advertise itself as the best route to all nodes (black hole effect) in an attempt to gain information.

Privacy can possibly be maintained using data encryption, access control, and restricting the network's ability to gather data at a sufficiently detailed level that could compromise privacy.

2.2.7 Secure Routing

The main challenge for secure routing is to ensure that each intermediate node cannot remove existing nodes or add extra nodes to the associated route. In the real world, a secure routing protocol guarantees the integrity, authenticity and availability of messages in the existence of adversaries. Every authorized receiver should receive all messages intended for it and should be capable of proving the integrity of these messages and also the identity of their sender. There are many routing protocols but they generally fail to

consider security in any serious manner. As discussed earlier, WSNs might be performing operations where they are dealing with sensitive data. Therefore given the insecure wireless communication medium, limited node capabilities, scalability, and possible insider threats, and that adversaries can use powerful laptops with high energy and long range communication capabilities to attack a network, designing a secure routing protocol is non-trivial [54].

Secure routing protocols for providing security from sources to destinations in WSNs must satisfy the following requirements [119]:

- Isolation of unauthorized nodes during the route discovery protocol.
- The network topology which depends on strong network bonds should not be revealed to an adversary.
- Security of the paths must be maintained. Otherwise an attacker is able to misdirect the network by advertising the false shortest path and possibly causing the formation of loops.
- Messages changed by an adversary and aberrant nodes can be identified.
- Unauthorised or aberrant nodes should not be able to change routing messages.

We will discuss all possible routing attacks in the section 2.3.5.

2.2.8 Secure Group Management

To manage a large scale network researchers generally recommend splitting the network into groups, clusters or domains, and also distributing the workload in an equitable way across these groups. Group management or cluster management protocols are used to maintain the groups of different nodes (adding or removing sensor nodes from a group, selecting/electing a new group leader, etc). Other services of the group management protocols help to increase network performance and consume fewer resources.

As we have described earlier, generally WSNs are assumed to be scalable. Therefore an energy efficient group management protocol is desirable. There exist different group management protocols. However these protocols have not considered security related

issues properly. For example, the in-network processing of raw data is performed in a WSN by dividing the network into small groups and analyzing the data aggregated at the group leaders. So a group leader has to authenticate the data it is receiving from other nodes in the group. This requires group key management. However, addition or deletion of nodes from the group leads to more problems. Moreover, these protocols need to be efficient in terms of energy, computation and communication to benefit WSNs. This means that traditional group management approaches are not directly implementable in WSNs due to their excessive memory and communication overheads [63]. Consequently, more cost-effective secure protocols for group management are needed.

2.2.9 Intrusion Detection

Intrusion detection is a type of security management system for computers and networks. An intrusion detection system (IDS) gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization). Intrusion detection functions include [64]:

- Monitoring and analysis of both user and system activities.
- Analysis of system configurations and vulnerabilities.
- Assessment of system and file integrity.
- Ability to recognize typical patterns of attacks.
- Analysis of abnormal activity patterns.
- Tracking of user policy violations.

According to Freiling et al. [65] an IDS for WSNs should satisfy the following properties:

- It must work with localized and partial audit data, as in WSNs there are no centralized points (apart from base stations and group leaders or cluster heads) that can collect global audit data. Thus this approach fits the WSN paradigm.

- It should utilize only a small amount of resources. A wireless network does not have stable connections, and physical resources of the network and devices, such as bandwidth and power, are limited. Disconnection can happen at any time. In addition, communication between nodes for intrusion detection purposes should not take too much of the available bandwidth.
- It cannot assume that any single node is secure. Unlike wired networks, sensor nodes can be very easily compromised. Therefore, in cooperative algorithms, the IDS must assume that *no* node can be fully trusted.
- It should be able to resist a hostile attack against itself. Compromising a monitoring node and controlling the behaviour of an embedded IDS agent should not enable an adversary to revoke a legitimate node from the network, or keep another intruder node undetected.
- The data collection and analysis should be performed at a number of locations and be truly distributed. The distributed approach also applies to the execution of detection algorithms and alert correlations.

We believe these requirements are reasonable in the context of WSNs IDS. However IDS is not the main focus of our research and we developed our own set of requirements mentioned in Chapter 4 that are more suitable for secure communication between source and destination.

2.2.10 Secure Data Aggregation

Most of a sensor node's energy is consumed during computation as well as sending and receiving of data packets. Sending one bit requires the same amount of energy as executing 50 to 150 instructions on sensor nodes [6]. Therefore reducing network traffic is important to save sensors' battery power in any WSN communication protocol.

To minimize the number of transmissions from thousands of sensor nodes towards a sink, a well known approach is to use in-network aggregation. The energy savings of performing in-network aggregation have been shown to be significant and are crucial for energy-constrained WSNs [57-59]. In a WSN sensed values should be transmitted to a sink but in many scenarios the sink does not need exact values from all sensors but rather

a derivative such as a sum, average or deviation. The idea of in-network aggregation is to aggregate the data required for the determination of the derivatives as closely to the data sources as possible instead of transmitting all sensed individual values through the entire network [56].

A serious issue connected with in-network data aggregation is data security [2]. Although previous works [57-60] do provide in-network data aggregation to reduce energy costs, these schemes assume that every node is honest which may not be suitable in terms of security. There are different types of attacks which can be harmful for in-network data aggregation, e.g., a compromised aggregator node or several compromised sensor nodes due to physical tempering could inject faulty data into the network. This will result in a corrupted aggregate. In many applications, nodes are communicating highly sensitive data, and due to such threats data privacy/security is vital. Aggregation becomes more challenging if end-to-end privacy between sensors and their associated sink is required [61-62].

2.3 Attacks on WSNs

Computer viruses, bugs and attacks have a history as long as computer networking itself. The first bug was identified in 1945. In 1960 the first threat to network security was identified; a white-collar crime performed by a programmer for the financial division of a large corporation. In 1983 Fred Cohen coined the term computer virus. One of the first PC viruses was created in 1986, called "The Brain". The history about computer and network security has been well documented [66-68]. Accordingly with improvements in the security of networks and computers we are now facing increasingly sophisticated attacks and threats.

In this section we will describe and discuss attacks and threats related to WSNs. Most of these attacks are similar to those that apply to traditional networks. However node captures are totally new and distinct attacks which do not apply to traditional networks. Further in this section we will describe attacks which are noxious and possibly lead towards a big damage in a network.

2.3.1 Node Capture Attacks

A node capture is one possible distinct attack on WSNs, where an adversary gains full control over a sensor node through direct physical access. The adversary can then easily extract cryptographic primitives and get unlimited access to the information stored on the node's memory chip, and can cause substantial damage to the entire system. This process can be done using reverse engineering followed by probing techniques that require access to the chip level components of the device [81]. It is usually assumed that node capture is easy, due to no physical restriction, to prevent access to sensor nodes in an outdoor environment [69].

Many researchers have proposed different key management schemes [72-80] for secure communication between sensor nodes. These schemes try to provide better resilience against node capture attacks, but still there is a chance of the entire network being compromised. For example in probabilistic key pre-distribution schemes [72, 75] compromising a few nodes can lead to the entire network communications being compromised.

One of our novel research contributions to provide high resilience against node capture attack has mentioned in section 1.7 and described in detail in section 5.5.1. Where we have discovered three main factors which can help adversaries during node capture attacks to compromise the communication of an entire sensor network [33, 71]:

- Node capture attacks can be a large threat if sensor nodes within the network share a key or keys with neighbouring nodes used to encrypt or decrypt data. Consequently the greater the level of key-sharing between neighbouring nodes the greater threat there is to communication privacy being compromised. Most existing solutions suffer from this drawback.
- The structure (topology) of a WSN affects the impact of node capture attacks. In general, the fewer the communication links between sensor nodes, the greater the possibility that an attacker can entirely block the communication paths between a source and a destination. For example, node capture attacks are generally more

effective in tree topologies than mesh topologies because in the former there is only one route from a child to its parent. If the parent node is compromised, the entire communication from its child nodes downward will potentially be compromised.

- The density of the WSN has a direct influence on node capture attacks, having a similar affect to the network structure. Consequently the optimum number of neighbouring nodes needs to be identified for specific applications after analysis. This has an effect on energy consumption, but more importantly when a sensor node with high density is physically captured this can lead towards compromise of larger sections of the WSN compared to a lower density sensor node.

By considering these three parameters we can improve resilience against node capture attacks. Key management schemes, which provide resilience against node capture attacks at the first level (i.e. pre-security), is not sufficient. Therefore post-security (i.e. second level security) solutions are also required to identify malicious or compromised sensor nodes, so that these compromised nodes can be excluded.

2.3.2 Side Channel Attacks

Side channel attacks are not part of this work (thesis). However we are discussing them in this section for the purposes of completeness.

Side channel attacks also fall into the category of physical tampering in the same way as node capture attacks. However this type of attack generally applies to all Wireless Networks. A side channel attack refers to any attack that is based on information gathered from the physical implementation of a cryptosystem, in contrast to a vulnerability in the algorithm itself [81]. For example the attacker monitors the power consumption or the Electro Magnetic (EM) emanation from such cryptographic devices, and then analyzes the collected data to extract the associated crypto key. These side channel attacks aim at vulnerabilities of implementations rather than algorithms, which make them particularly powerful since adversaries are not required to know the design of the target system. Simple Power Analysis (SPA), Differential Power Analysis (DPA), Simple

Electromagnetic Analysis (SEMA), and Differential Electromagnetic Analysis (DEMA) are side channel attacks that enable extraction of a secret key stored in cryptographic devices [83].

Simple power analysis [70] is a technique that involves directly interpreting power consumption measurements collected during cryptographic operations. No statistical analysis is required in such an attack. The analysis can yield information about a device's operation as well as key material. It can be used to break cryptographic implementations in which the execution path depends on the data being processed.

Similarly, in simple electromagnetic analysis [84], an adversary is able to extract compromising information from a single electromagnetic sample.

In differential power analysis [70], an adversary monitors the power consumed by cryptographic devices, and then statistically analyzes the collected data to extract a key in contrast to the simple power analysis.

In differential electromagnetic analysis [84], instead of monitoring the power consumption, an attacker monitors electromagnetic emanations from cryptographic devices, and then the same statistical analysis as that for the differential power analysis is performed on the collected electromagnetic data to extract secret parameters [81].

Side channel attacks are also possible in WSNs. Okeya et al. describes the fact that a side-channel attack on Message Authentication Codes (MAC), using simple Power Analysis as well as Differential Power Analysis, is possible in WSNs [82]. Their results suggest that several key bits can be extracted through the power analysis attack. This leads to the conclusion that protecting block ciphers against side channel attacks is not sufficient. Further research is required to explore all possible security measures for Message Authentication Codes as well.

TinySec is link layer security architecture for WSNs, and it uses a block cipher encryption scheme for its implementation. According to the previous discussion, such an encryption scheme shows a weakness of the TinySec protocol [85].

Additionally, timing attacks come under the category of side-channel attacks. They have not yet been explored in the context of WSNs. A timing attack makes use of algorithms

which have non-constant execution times and can potentially leak secret information. Non-constant execution times can be caused by conditional branching and various optimization techniques. The operating system running on sensor nodes is event-driven and extremely optimized in terms of memory consumption. This suggests that a timing side-channel attack is possible. A solution to this type of attack is to use constant execution time software. However, it is not clear if this is easily achievable in WSNs. Therefore, searching for countermeasures for timing attacks in WSNs is an important area for future research.

Some countermeasures for side-channel attacks used in traditional and embedded systems are [81]:

- power consumption randomization,
- randomization of the execution of the instruction set,
- randomization of the usage of register memory,
- CPU clock randomization,
- using fake instructions,
- using bit splitting.

2.3.3 Denial of Service (DoS)

A denial of service attack is any event that diminishes or eliminates a network's capacity to perform its expected function through hardware failures, software bugs, resource exhaustion, malicious broadcasting of high energy signals, environmental conditions, or any complicated interaction between these factors. Communication systems could be jammed completely if such attacks succeed. Other denial of service attacks are also possible, e.g. inhibiting communication by violating the MAC protocol.

One of the standard protections against jamming utilizes spread spectrum communication. However, cryptographically secure spread spectrum radios are not available commercially. Also, this protection is not secure against adversaries who can capture nodes and remove their cryptographic keys [86].

Each protocol layer in a WSN is defenceless to different DoS attacks and has different options available for its defence. Some of the attacks crosscut multiple layers or exploit interactions between them. For example, at the network layer in a homing attack, the attacker looks at network traffic to deduce the geographic location of critical nodes, such as cluster heads or neighbours of a base station [29]. Furthermore in the routing and network layer, due to a "misdirection" attack, messages could flood the network. This could also happen by looking at the routing table or negative advertising by the adversary to flood either a sender, receiver or an arbitrary node [120]. Table 2-1 shows a typical sensor's network layers and describes each layer's vulnerabilities and defences [87].

Sensor network layers and denial-of-services defence		
Protocol layer	**Attacks**	**Defences**
Physical	Jamming	Spread-spectrum, priority message, lower duty cycle, region mapping, mode change
	Tampering	Tamper-proofing, hiding
Link	Collision	Error-correction code
	Exhaustion	Rate limitation
	Unfairness	Small frames
Network and Routing	Neglect and greed	Redundancy, probing
	Homing	Encryption
	Misdirection	Egress filtering, authentication, monitoring
	Black holes	Authorization, monitoring, redundancy
Transport	Flooding	Client puzzles
	De-synchronization	Authentication

Table 2-1: Sensor protocol layers and DoS defences.

According to Wood et al., every DoS attack is perpetrated by someone [88]. The attacker has an identity and a motive, and is able to do certain things in or to a WSN. An attack targets some service or layer by exploiting some vulnerability. An attack may be

thwarted, or it may succeed with varying results. Each of these elements is necessary for understanding the whole process of a DoS attack. Any useful and intuitive DoS taxonomy should answer the following questions:

- Who is the attacker?
- What is she/he capable of?
- What is the target?
- How is it attacked?
- What are the results?

Wood et al. [88] also answer each question listed above in turn. Taken together the attacker, capability, target, vulnerability, and results describe a DoS attack against a WSN.

2.3.4 Software Attacks

Software-based attacks in WSNs can also be dangerous. For this type of attack, an adversary may try to modify the code in memory or exploit known vulnerabilities in the code. A well-known example of such an attack is a *buffer overflow* attack. Buffer overflow refers to the scenario where a process attempts to store data beyond the boundaries of a fixed length buffer. This results in the extra data overwriting the adjacent memory locations [81].

Such attacks can easily apply to TinyOS - an operating system developed for sensor nodes with limited resources. The current implementation of TinyOS does not provide any memory access control, i.e. there is no function to control which users/processes access which resources on the system, and what type of execution rights they have. In TinyOS the assumption is largely that a single application or user controls the system [81]. However in traditional operating systems, access control involves authenticating processes, and then mediating their access to different system resources.

Regehr et al. have presented the concept of drawing a *red line*, which refers to having a boundary between trusted and un-trusted code. Their solution, called Un-trusted Extension for TinyOS (UTOS), uses a concept similar to sandboxing. This solution

provides an environment in which un-trusted, and possibly malicious, code could be run without affecting the kernel [97, 98].

Similarly TinyOS uses the concept of Active Messaging (AM). AM is an environment that facilitates message-based communication in distributed computer systems. Each AM message consists of the name of a user-level handler on the target node that needs to be invoked as well as the data that needs to be passed on [99]. This approach enables the implementation of a TCP/IP-like network stack on the sensor node that fits the hardware limitations of the sensor nodes. Roosta et al. have pointed out another weakness in TinyOS, resulting from port operations. It is possible to open a port to a remote sensor node using a USB port and a PC. The *serial forwarder*, which is one of the most fundamental components of TinyOS software, can be called to open a port to a node. There is no security check to authenticate the user who is attempting to open the port. This could lead to an attack on the software whereby an adversary opens a port to the node and uploads software, or downloads information from the node [81].

The following countermeasure can be considered to secure the TinyOS software and protect the software from being exploited by malicious users:

- Software authentication and validation, e.g. remote software-based attestation for sensor networks [100].
- Defining accurate trust boundaries for different components and users.
- Using a restricted environment such as the Java Virtual Machine.
- Dynamic run-time encryption/decryption for software: this is similar to the encryption/decryption of data except that the code running on the device is encrypted. This can prevent a malicious user from exploiting the software [81].
- Hardware attestation. The trusted computing group platform and next generation secure computing base provide this type of attestation [101]. A similar model could be used in sensor networks.

2.3.5 Routing Attacks

As described earlier in WSNs, every node acts as a router. Routing and data forwarding is an important task for sensor nodes. Routing protocols have to be energy and memory efficient but at the same time they have to be robust against attacks and node failures. There have been many power-efficient routing protocols proposed for WSNs. However, most of them suffer from different security vulnerabilities. In the real world, a secure routing protocol should guarantee the integrity, authenticity and availability of messages in the existence of adversaries of arbitrary power. Every authorized receiver should receive all messages proposed for it and would be capable of proving the integrity of every message and also the identity of the sender [54]. We briefly describe a few attacks on routing protocols:

- ***Black hole attack or packet drop attack:*** An attacker can drop received routing messages, instead of relaying them as the protocol requires, in order to reduce the quantity of routing information available to other nodes. This is called a *black hole attack* [102]. This attack can be launched selectively (dropping routing packets for a specified destination, a packet every *t* seconds, or a randomly selected portion of each packet) or in bulk (drop all packets), and may have the effect of making the destination node unreachable or downgrade communications in the network [103].
- ***Spoofed, altered, or replayed attack*:** In this attack an adversary can record old valid control messages and re-send them, causing the receiver node to lose energy quickly. As the topology changes, old control messages, though valid in the past, may describe a topology configuration that no longer exists. An attacker can perform a replay attack to make other nodes update their routing tables with stale routes. This attack can be successful even if control messages bear a digest or a digital signature that does not include a timestamp [102].
- ***Wormholes attack:*** This attack [104] is quite severe, and consists in recording traffic from one region of the network and replaying it in a different region. This attack is particularly challenging to deal with since the adversary does not need to compromise any nodes and can use laptops or other wireless devices to send

packets on a low latency channel. Hu et al. [104] proposed the concept of packet leashes where additional information is added to a packet, the purpose of which is to restrict the maximum distance the packet can travel in a given amount of time [81].

- ***Selective forwarding attack:*** In this attack, a malicious node selectively drops sensitive packets. Selective forwarding attacks are typically most effective when the attacking nodes are explicitly included on the path of a data flow. Yu et. al. [105] proposed a light weight detection scheme which uses a multi-hop acknowledgement technique to launch alarms by obtaining responses from intermediate nodes.
- ***Sinkhole attack:*** In this attack, an adversary tries to attract as much traffic as possible toward compromised nodes. The impact of the sinkhole is that it can be used to launch further active attacks on the traffic that is routed through it. The severity of active attacks increases multi-fold especially when these are carried out in collusion [106]. Sinkhole attacks typically work by making a compromised node look especially attractive to surrounding nodes with respect to the routing algorithm. For instance, an adversary could spoof or replay an advertisement for an extremely high quality route to a sink [41].
- ***HELLO flood attack:*** The preference for the shortest communication route can usually be exploited by a HELLO flood. In the case of multi-hops, this means broadcasting a message with a long-range radio-antenna to all nodes in the network, stating the node performing the HELLO flood is the base station. The receiving nodes should then conclude that the route through the node sending the HELLO flood is the shortest. They will try to send all their succeeding messages through this node, which most probably is not even within radio range. In the worst case, all nodes in the network will keep sending their messages into oblivion. An attack such as the HELLO flood is meant to completely disable the WSN and prevent it from performing its tasks [107].
- ***Acknowledgement spoofing:*** The goal of an adversary in this attack is to spoof a bad link or a dead node using the link layer acknowledgement for the packets it overhears for those nodes.

2.3.6 Traffic Analysis Attacks

In WSNs all communication is moving toward a sink or base station in many-to-one or many-to-few patterns. An adversary is able to gather a lot of information on the topology of the network as well as the location of the base station and other strategic nodes by observing the traffic volume and pattern [81].

Deng *et al.* have defined two types of traffic analysis attacks in WSNs: a *rate monitoring* attack and a *time correlation* attack. In a *rate monitoring* attack, an adversary monitors the packet sending rate of nodes near the adversary, and moves closer to the nodes that have a higher packet sending rate. In a *time correlation* attack, an adversary observes the correlation in sending time between a node and its neighbour node that is assumed to be forwarding the same packet, and deduces the path by following the "sound" of each forwarding operation as the packet propagates towards the base station [109].

The possible solutions to the traffic analysis attacks are to use randomness and multiple paths in routing, using probabilistic routing and the introduction of fake messages in the network. In the case of fake messages it can increase the communication overhead. Therefore it might not be a cost effective solution.

2.3.7 Sybil Attack

The Sybil attack is defined as a "malicious device illegitimately taking on multiple identities" [101]. For example, a malicious node can claim false identities, or impersonate other legitimate nodes in the network [81]. Perrig et al. have pointed out that the Sybil attack can affect a number of different protocols [101]:

- Distributed Storage Protocols.
- Routing Protocols.
- Data Aggregation (used in query protocols).
- Voting (used in many trust schemes).
- Fair Resource Allocation Protocols.

- Misbehaviour Detection Protocols.

To attack the routing protocols, the Sybil attack would rely on a malicious node taking on the identity of multiple nodes, and thus routing multiple paths through a single malicious node [41]. However the Sybil attack can operate in different orders to attack the protocols listed above.

The proposed solutions to the Sybil attack include: 1) radio resource testing which relies on the assumption that each physical device has only one radio, 2) random key predistribution which associates the identity of each node to the keys assigned to it and validates the keys to establish whether the node is really who it claims to be, 3) registration of the node identities at a central base station, and 4) position verification which makes the assumption that the WSN topology is static.

2.3.8 Attacks on In-network Processing

In-network processing, also called data aggregation, was discussed in terms of secure data aggregation in section 2.2.10. Data aggregation is very useful in terms of reducing the communication overhead. However there can be different types of attack on in-network processing:

- Compromise a node physically to affect aggregated results [117].
- Attack aggregator nodes using different attacks.
- Send false information to affect the aggregation results [164].

To handle these possible attacks, there should be an efficient security solution to stop the adversary from affecting aggregated results. Furthermore this security solution should be capable of providing resilience against attacks (routing attacks etc.) on aggregator nodes. It is necessary to have mechanisms to provide accurate information to end users after successful attacks on aggregator nodes or results.

2.3.9 Attacks on Time Synchronization Protocols

Time synchronization protocols provide a mechanism for synchronizing the local clocks of nodes in a WSN. There are various different protocols proposed for time synchronization. Three of the most prominent protocols are the Reference Broadcast Synchronization (RBS) [6], Timing-sync Protocol for Sensor Networks (TPSN) [9], and Flooding Time Synchronization Protocol (FTSP) [20].

Most of the time synchronization protocols don't consider security. An adversary can easily attack any of these time synchronization protocols by physically capturing a fraction of the nodes and injecting them with faulty time synchronization message updates. In effect, this makes the nodes in the entire network out-of-sync with each other. Time-synchronization attacks can have a significant effect on a set of WSN applications and services since they heavily rely on accurate time synchronization to perform their respective functions [81].

2.4 Security in Mobile Sensor Networks

Secure communication between network components is always an issue, and researchers are continually inventing new security protocols to provide more and more secure communications. Although security has long been an active research topic in traditional networks, the unique characteristics of MSNs present a new set of nontrivial challenges to security design. These challenges include the open network architecture, shared wireless medium, resource constraints, scalability, and highly dynamic network topologies of MSNs. Consequently, the existing security solutions for traditional networks, mobile ad hoc networks and static sensor networks do not directly apply to MSNs [36].

The ultimate goal of security solutions for MSNs is to provide security services, such as authentication, confidentiality, integrity, anonymity and availability, to mobile nodes.

Node mobility poses far more dynamics in MSNs compared to SSNs (Static Sensor Networks). The network topology is highly dynamic as nodes frequently join and leave the network, and roam in the network. The wireless channel is also subject to

interferences and errors, revealing volatile characteristics in terms of bandwidth and delay. The dynamics nature of MSNs increases security challenges, as mobile nodes may request for anytime, anywhere security services as they move from one place to another.

2.5 Summary

This chapter starts with general (non-security) challenges in WSNs, which include fault tolerance, topologies, routing, mobility and scalability. We have then discussed in detail about security challenges in WSNs, including data confidentiality, data integrity, authentication, key establishment, availability, privacy, secure routing, secure group management, intrusion detection and secure data aggregation. Finally we have described possible attacks on WSNs, which include node capture attacks, side channel attacks, denial of service attacks, software attacks, routing attacks, traffic analysis attacks, Sybil attacks and attacks on in-network processing. Some of these attacks are similar to those in traditional networks, e.g. routing attacks and DoS attacks, while the others only exist in WSNs. In particular, node capture attacks may allow an adversary to compromise the security of an entire WSN. Therefore this attack is the centre of attention of many researchers. The uniqueness of the node capture attack and its after attack effects on WSN is more challenging. The current work is using different key establishment techniques to reduce the after damage in case some sensor nodes are physically compromised. However due to the possibility of physically compromising sensor nodes data confidentiality is at high risk. The compromised sensor node could be an aggregator or group leader sensor node. All these issues are interlinked with each other. Therefore we require that key management solution should provide high resilience against node capture attack using less resource, efficient secure data aggregation to achieve better confidentiality for different applications, secure and efficient group leader/aggregator selection scheme and key management for mobile sensor node in case nodes roam from one place to another.

In the next chapter, we will present a critical survey on current key management, secure data aggregations, secure group leader election and key management for MSN protocols, and also discuss their specific limitations against the requirements of WSNs.

Chapter Three: Key Management in WSNs

Key establishment/management is the basic technique to achieve data confidentiality, data integrity, authenticity, secure data routing and secure data aggregation. Due to the importance of key establishment/management in order to achieve information security we therefore describe existing WSNs key management solutions in the start of this chapter. For a better understanding we have divided these existing key management solutions into five different categories; key pool based key management, session based key management, hierarchical based key management, group based key management and key management for heterogeneous sensor networks. Furthermore secure data aggregation is presented in section 3.2, replication attacks described in section 3.3, secure group leader election/selection discussed in section 3.4 and key management for MSNs is presented in section 3.5.

3.1 Key Management in WSNs

When setting up a WSN, one of the initial requirements is to establish cryptographic keys for later use. In indoor and outdoor WSN applications, communications can be monitored and nodes are potentially subject to capture and surreptitious use by an adversary [72]. For this reason cryptographically protected communications are required. A keying relationship can be used to facilitate cryptographic techniques. Cryptographic techniques are categorized as either symmetric or asymmetric forms of cryptography. Symmetric cryptography relies on a shared secret key between two parties to enable secure communication. Asymmetric cryptography, on the other hand, employs two different keys, a private one and a public one. The public key is used for encryption and can be published. The private key is used for decryption. From a computational point of view, asymmetric cryptography requires orders of magnitude more resources than symmetric cryptography and similarly a public key infrastructure (PKI) would be required, which may be difficult to achieve in the ad-hoc environment of a WSN. Therefore, recently only symmetric cryptosystems have been proposed and recommended for WSNs.

An ideal key management solution for WSN would satisfy the design criteria of providing secure communication between source and destination, scalability and resource efficiency. There are two simple strategies for symmetric key management schemes for WSNs. One is to use a single secret key over the entire WSN. This scheme is obviously efficient in terms of the cost of computation and memory. However the compromise of only a single sensor node exposes all communications over the entire WSN, which is a serious deficiency. The other approach is to use distinct keys for all possible pairs of sensor nodes. Then every sensor node is preloaded with $n - 1$ keys, where n is the WSN size. This scheme guarantees perfect resilience in that links between non-compromised sensor nodes are secure against any coalition of compromised sensor nodes. However this scheme is not suitable for large-scale WSNs since the key storage required per sensor node increases linearly with the WSN size [78]. If there is a network of 10,000 sensor nodes, then each node must store 9999 keys in their memory. Since sensor nodes are resource-constrained, this significant overhead limits the scheme's applicability, but it can be effectively used for smaller WSNs. Consequently, in the first strategy the sharing of keys between sensor nodes is high whilst in the second strategy sharing between the sensor nodes is low. Due to the need for secure communication with only limited resources, researchers are proposing solutions that fall between these two strategies.

3.1.1 Key pool Based Key Management

3.1.1.1 Random key predistribution scheme (Basic Scheme)

Eschenacuer and Gilger [72] proposed a probabilistic key pre-distribution scheme. This scheme is also known as the Basic Scheme. This scheme is divided into three parts: key pre-distribution, shared-key discovery, and path key establishment [144].

(a) Key predistribution phase: There is a large key pool S of [S] keys with unique identifiers. Every sensor node is equipped with a fixed number of keys randomly selected from this key pool with their key identifiers. Once keys and their identifiers are assigned

to every sensor node in the WSN, trusted nodes will be selected as controller nodes, and all the key identifiers and their associated sensor identifiers will be saved on the controller nodes. These few keys are enough to ensure that any two nodes share a common key, possibly through the assistance of other nodes, based on a selected probability.

(b) Shared-key discovery phase: Once nodes are successfully deployed in a target application, every pair of nodes within their wireless communication range establishes a common keys. If they share any common key(s) among their assigned keys, they can pick one of them as their shared secret key. There are many ways for finding out whether two nodes share common keys or not. The simplest way is to make the nodes broadcast their key identifier lists to other nodes. If a node finds out that it shares a common key with a particular node, it can use this key for secure communication. This approach does not give an adversary any new attack opportunities and only leaves some room for launching a traffic analysis attack in the absence of key identifiers.

(c) Path key establishment phase: As discussed earlier communication can be established between two sensor nodes only if they share a key, but the path key establishment stage facilitates provision of a link between two sensor nodes when they do not share a common key. Let us assume that a sensor node x wants to communicate with another sensor node y, but they do not share a common key between them. x can send a message to a sensor node u, saying that it wants to communicate with y, where the message is encrypted using the common key shared between x and u. If u has a key in common with y, it can generate a pair-wise key K_{xy} for x and y, thereby acting like a key distribution centre or a mediator between x and y. All the communications are in an encrypted form using their respective shared keys.

The advantages of this scheme include the fact that it is flexible, efficient, and fairly simple to employ. The disadvantages of this scheme include that it cannot be used in circumstances demanding heightened security or node to node authentication, and it provides only limited scalability. Compromise of a controller sensor node and certain number of other sensor nodes can lead the adversary to compromise the entire WSN [144].

3.1.1.2 Q-composite random key predistribution scheme

Chan et al. [75] extended the previous idea of the Basic Scheme [72] to overcome the difficulties that occur when a pair of sensor nodes share no common key. Chan et al. proposed two different variations of the Basic scheme: *Q-Composite Random Key Predistribution* and *Multipath Key Reinforcement*, and a variation of the commonly known Pairwise Scheme, called the *Random Pairwise Scheme*. In the Basic Scheme [72], two nodes share a unique key for establishing secure communications. A given network's resilience to node capture can be improved by increasing the number of common keys that are needed for link establishment. The Q-Composite Random Key Predistribution Scheme does this by requiring that two nodes have at least q common keys to set up a link [75]. As the amount of key overlap between two sensor nodes is increased, it becomes harder for an adversary to compromise their communication link. At the same time, to maintain the probability that two sensor nodes establish a link with q common keys, it is necessary to reduce the size [S] of the key pool S, which poses a possible security breach in the network as the adversary now has to compromise only a few nodes to gain a large part of S. So the challenge of the Q-Composite Scheme is to choose an optimal value for q while ensuring that security is not sacrificed [144].

The first phase of the Basic and Q-composite schemes are the same but in the second phase these two schemes differ in that the Q-Composite Scheme requires each node to identify neighbouring sensor nodes with which they share at least q common keys, while the Basic scheme only requires one shared key. This restriction in the Q-Composite Scheme allows the number of keys shared to be more than q but not less. At this stage in the process, nodes will fail to establish a link if the number of keys shared is less than q; and otherwise they will form a new communication link using the hash of all the q keys as a shared key, denoted as $K = \text{hash}(k_1|k_2|. . . |k_q)$ where | is used for concatenation. The size of the key pool S is an important parameter that needs to be calculated. The Q-composite scheme provides better resilience against node capture attacks. The amount of communications that are compromised in a given network with the Q-Composite Scheme applied is 4.74 percent when there are 50 compromised nodes, while the same network

with the Basic scheme applied will have 9.52 percent of communications compromised [144]. Though the Q-Composite Scheme performs badly when more sensor nodes are captured in a WSN, this may prove a reasonable concession as adversaries are more likely to commit a small-scale attack and preventing smaller attacks can push an adversary to launch a large-scale attack, which is far easier to detect.

The advantages of the Q-Composite scheme include that it provides better security than the Basic Scheme by requiring more keys to share with neighbouring sensor nodes for communication, which makes it difficult for an adversary to compromise the communication of a sensor node. The disadvantages of this scheme include that it is vulnerable to breakdown under large-scale attacks, and does not satisfy scalability requirements.

Furthermore, the multipath key reinforcement scheme [75] provides good security with additional communication overhead. In previous schemes there is an issue that the links formed between sensor nodes after the key discovery phase may not be totally secure due to the random selection of keys from the key pool, allowing some sensor nodes in a WSN to share the same keys. This could threaten the security of these sensor nodes when only one of them is compromised.

To solve this problem, the communication keys must be updated when a sensor node is compromised. This should not be done using the old established links, as an adversary would then be able to decrypt the communications to obtain new keys. Instead it should be coordinated using multiple independent paths for greater security.

The advantages of this scheme include that it offers better security than the Basic scheme or the Q-Composite Scheme. The disadvantages of this scheme include that it creates communication overhead that can lead to depleted node battery life and the chance for an adversary to launch a DoS attack.

3.1.1.3 Polynomial pool-based key pre-distribution

Liu et al. [128] designed two schemes for secure pair-wise communication in Wireless Sensor Networks: *Polynomial-based* and *grid-based* key distribution protocols. The

polynomial-based protocol further extends the idea of Eschenauer et al. works [72]. Instead of pre-distributing keys, they actually pre-distribute *polynomials* from a *polynomial pool.* This polynomial based key pre-distribution scheme offers several efficient features compared to other schemes:

- Any two sensor nodes can definitely establish a pair-wise key when there are no compromised sensors.
- Even with some sensor nodes being compromised, the others in the WSN can still establish pair-wise keys.
- A node can find the common keys to determine whether or not it can establish a pairwise key and thereby help reduce communication overhead [144].

The drawback of this scheme is that compromising more than t polynomials leads to sensor network compromise. Further to avoid such attacks each node must store 2 bivariate t-degree polynomials and IDs of the compromised nodes, which is resulting in additional memory overhead.

3.1.1.4 Hypercube key distribution scheme

The *Hypercube Key Distribution Scheme* [125] guarantees that any two sensor nodes in the WSN can establish a pair-wise key if there are no compromised sensor nodes present as long as the two sensor nodes can communicate. Also, sensor nodes can still communicate with a high probability even if compromised sensor nodes are present. Sensor nodes can decide whether or not they can directly communicate with other sensor nodes and what polynomial they should use when transmitting messages. If two sensor nodes do not share a common polynomial, they have to use a path discovery method to compute an indirect key.

The path discovery algorithm described by Ning et al. [125] finds paths between a pair of sensor nodes a and b dynamically. In this method, the source and other sensor nodes communicate with a sensor node that is uncompromised and has a closer match to the destination sensor node compared to the Hamming distance of their IDs where the

Hamming distance is defined as a measure of the difference between two binary sequences of equal length. If there are no compromised sensor nodes in the WSN, this scheme will always work as long as any two sensor nodes can communicate.

There are a number of attacks that can be applied to the current scheme. One attack is to attempt to compromise the polynomials used in key generation between sensor nodes *a* and *b* without compromising the sensor nodes themselves. To achieve this, the attacker must first compromise $t + 1$ other sensor nodes. If the sensor nodes *a* and *b* have computed an indirect key, the attacker must compromise the sensor nodes used in the path that established the key. In total, the attacker must compromise $n \times (t + 1)$ (where n is the number of polynomials and t is number of compromised node IDs) sensor nodes to effectively prevent sensor nodes *a* and *b* from communicating with each other. A second attack against the scheme is to damage the whole WSN. One way to do this is to compromise a number, *b*, of polynomials distributed to the sensor nodes in the WSN. This will affect the indirect keys computed. A further way to attack the WSN as a whole is to randomly compromise individual sensor nodes. This could compromise the path discovery process and make it more expensive to create pair-wise keys [125, 144].

3.1.1.5 Key management schemes using deployment knowledge

Du et al. [78] propose a scheme using deployment knowledge that is based on the Basic Scheme [72]. Deployment knowledge in this scheme is modelled using probability density functions (PDFs). All the schemes discussed until now considered the PDF to be uniform, so knowledge about sensor nodes cannot be derived from it. Du et al. consider non-uniform PDFs, which means that they assume the positions of sensor nodes to be in certain areas. Their method first models sensor node deployment knowledge in a WSN and then develops a key pre-distribution scheme based on this model.

As in the Basic Scheme, the key pre-distribution scheme also consists of three phases for the deployment model: key pre-distribution, shared key discovery, and path key establishment. This scheme differs only in the first stage while the other two stages are similar to those of the Basic Scheme.

Key pre-distribution: In this phase the scheme divides the key pool KP into $t \times n$ key pools $KP_{i,j}$ of size $\omega_{i,j}$. The goal of dividing the key pool is to ensure that neighbouring key pools have more keys in common. The pool $KP_{i,j}$ is used for the nodes in the group $G_{i,j}$. Given $\omega_{i,j}$ and overlapping factors α and β, the key-pool is divided into subsets so that (i) two horizontally and vertically neighbouring key-pools have $\alpha \times \omega_{i,j}$ keys in common, (ii) two diagonally neighbouring key-pools have $\beta \times \omega_{i,j}$ keys in common, and (iii) non-neighbouring key-pools do not share a key. Two key pools are neighbours if their deployment groups have nearby resident points (x_i, y_j) for $1 < i < t$ and $1 < j < n$, where the points are arranged in a two dimensional grid. After the key pool is divided, each node in a group $G_{i,j}$ is selected and keys are installed from the corresponding subset key pools. As mentioned earlier, for the current scheme the *Shared discovery phase* and *Path key establishment phase* are exactly the same as for the Basic Scheme [72] described in section 3.1.1.1.

According to Du *et al.* [78] an increase in the number of random keys chosen from the key pool for each sensor node will increase the connectivity, which is true. Moreover they show that if we can carry 100 keys in each sensor node using their method the probability of local connectivity with neighbouring nodes will be 0.687. Now, suppose C_n is the number of compromised nodes and m is the number of compromised keys. The compromise of more nodes will allow an adversary to get more keys. Suppose we have a network of 10,000 sensor nodes. If an adversary gets 10 keys then the probability that it can communicate with any other node will be 0.024. If the number of compromised keys increases to 120 through the compromise of C_n nodes, the probability will increase to 0.871. We represented this in the graph shown in Figure 3-1 below. This graph shows that the compromise of more nodes will help to compromise the complete network.

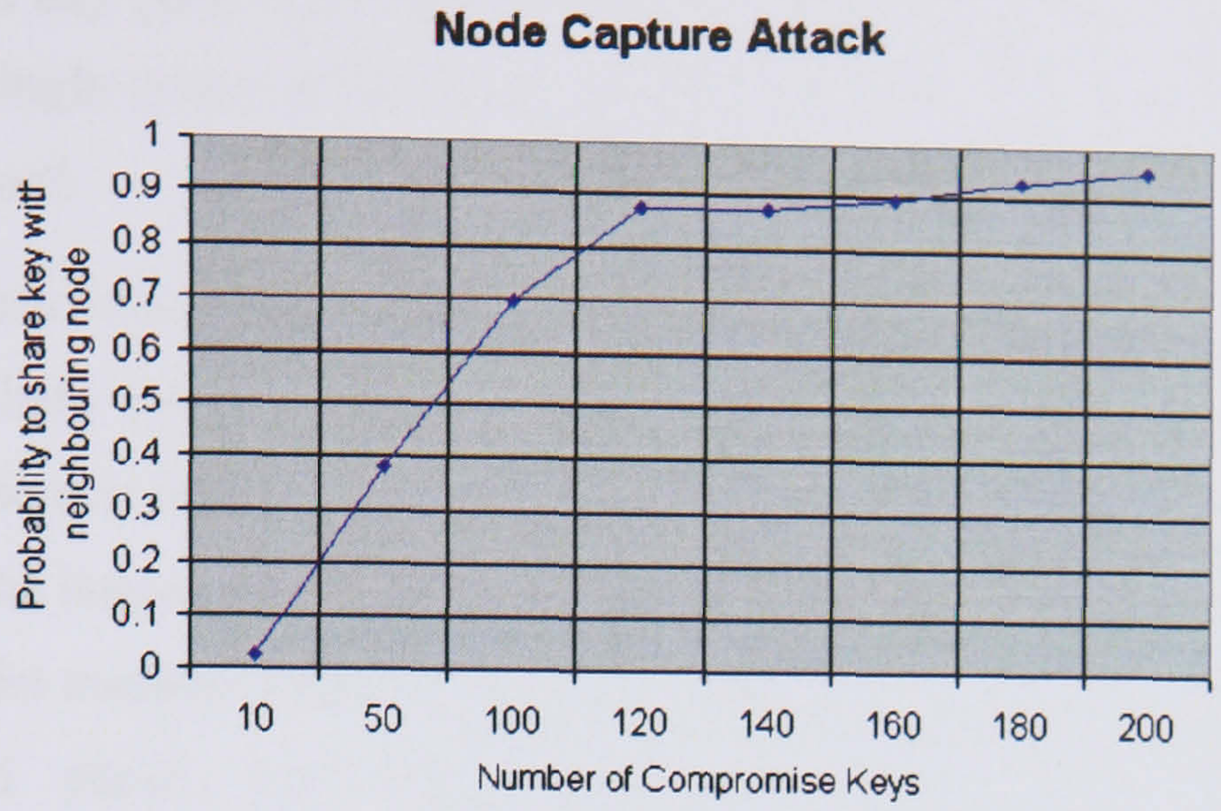


Figure 3-1: Compromise of a sensor network using node capture attacks on the scheme of Du et al. [78].

The advantages of this scheme include the fact that, by only considering deployment knowledge that can minimize the number of keys and help to reduce network overhead, it increases overall connectivity of the network graph, and offers the same benefits over the Basic Scheme on which it is based. The problem in this scheme is the difficulty and complexity in deciding the parameters $\omega_{i,j}$, α and β to provide adequate key connectivity.

3.1.1.6 Location dependent key management scheme

The location dependent key management scheme proposed by Anjum [126] decides which keys to put on each node depending on their locations in the environment. In this scheme, nodes are determined to be static. They communicate only through encrypted channels and nodes can be added at any time. Also nodes in this scheme are assumed to be capable of transmitting at different power levels and giving different transmission ranges. Also there exist special nodes called anchors. The only difference between the anchor nodes and the other nodes in the network is that the anchor nodes transmit at different power levels and are tamper proof. There are also three phases in the scheme: a pre-distribution phase, an initialization phase, and a communication phase. In the pre-distribution phase, a key server computes a set of keys to be used by the nodes. It places

the keys into a key pool. Each sensor node is then loaded with a subset of these keys along with a single common key every node shares. The anchor nodes do not get keys from the key pool.

All of the nodes and anchors are randomly distributed. During the initialization phase the anchor nodes help the other sensor nodes to change their existing keys by providing beacons. The sensor nodes receive these beacons and compute new keys based on their old keys and the beacons received from the anchor nodes. The original subset of keys is deleted from the memory of the sensor nodes after they compute their new keys. In the communication phase, the nodes compute pair-wise keys to establish secure communication among them. One of the significant advantages of this location-aware key management scheme is that compromised nodes do not affect nodes in a different location of the network.

This scheme also performs worse than a random key distribution scheme with a key pool size of 5000 and 175 keys on each sensor node. As the numbers of compromised sensor nodes in the WSN are increased, the performance of the random key distribution scheme deteriorated faster than the location-dependent scheme [126]. Furthermore anchor nodes create an extra overhead on the WSN.

In the location dependent key management scheme, an adversary can launch a denial of service attack if they jam the anchor nodes and transmit false beacons. This is fairly hard to accomplish since anchor nodes are randomly dispersed in the environment. There is no alternative when anchor nodes are physically compromised.

According to Zhou et al., random key pre-distribution schemes suffer from two major problems, making them inappropriate for many applications. First these schemes require that the deployment density is high enough to ensure connectivity. Second the compromise of a set of keys or key spaces leads toward compromise of the entire WSN [80].

PIKE [76] addresses the problem of the high density requirement of random key pre-distribution schemes [122]. In PIKE, each sensor node is equipped with an ID of the form (i,j), corresponding to a location on a $\sqrt{n} \times \sqrt{n}$ grid, where n is the network size. Each sensor is also preloaded with a number of pair-wise keys, each of which is shared with a

sensor that corresponds to a location on the same row or the same column of the grid. Now any pair of sensors that does not share a preloaded pair-wise key can use one or more peer sensors as trusted intermediaries to establish a path key. PIKE requires network-wide communications to establish path keys, each of which requires $O(\sqrt{n})$ communication overhead. This is a relatively high communication overhead, making it unsuitable for large WSNs.

3.1.2 Session Based Key Management

3.1.2.1 SPINS

A number of shared-session key negotiation protocols have been developed for WSNs. *SPINS* [39] is a security suite that includes two protocols: *SNEP* and *μTESLA*. SNEP is for confidentiality, two party data authentication, integrity and data freshness, whilst μTESLA provides authentication for data broadcasting. Suppose that a node x wants to establish a shared session key SK_{xy} with another node y through a trusted third party sink S. The sink plays a role as the key distribution centre. x will send a request message to y. y receives this message and sends a message to S. S will perform the authentication and generate the shared session key and send this key back to x and y respectively.

Liu et al., [44] quote in their paper that μTESLA [39] will not be efficient in large WSNs. For example, let μTESLA use 10Kbps bandwidth and support 30 byte messages. To bootstrap 2000 sensor nodes, the sink has to send or receive at least 4000 packets to distribute the initial parameters, which takes at least $4000 \times 30 \times 8 / 10240 = 93.75$ seconds even if the channel utilization is perfect. Such a method certainly cannot scale up to very large WSNs, which may have tens of thousands of sensor nodes.

Therefore multi-level μTESLA schemes have been proposed to extend the capability of the original μTESLA protocol [20, 21]. An improved version of the μTESLA system uses broadcasting of the key chain commitments rather than μTESLA's unicasting technique. They present a series of schemes starting with a simple pre-determination of key chains and finally settling on a multi-level key chain technique.

Liu et al. [43] have found weaknesses in their own work [44, 45] and suggest that these issues which are not properly addressed in their publications [44, 45] need to be addressed. These are described below:

DoS attacks: The multi-level μTESLA schemes scale broadcast authentication up to large networks by constructing multi-level key chains and distributing initial parameters of lower-level μTESLA instances with higher-level ones. However, multi-level μTESLA schemes magnify the threat of DoS attacks. An attacker may launch DoS attacks on the messages carrying the initial μTESLA parameters [44, 45]. Though several solutions have been proposed in Liu et al. [44], they either use substantial bandwidth or require significant resources to be available to senders [43].

3.1.2.2 BROSK

The *BROadcast Session Key* (BROSK) [129] negotiation protocol stores a single master key in each sensor node for the entire WSN. A pair of sensor nodes (S_i, S_j) exchange random nonce values N_i and N_j. The master key K_m is used to establish a session key K_{ij} = MAC (K_m | N_i | N_j), where "|" is used for concatenation and MAC is a Message Authentication Code function [115]. There are a couple of issues which are not described by BROSK. If the master key is compromised after a node capture attack, an adversary can easily compromise the entire network communication and generate all other keys. The BROSK protocol also didn't present the effect of node capture attacks in their scheme, when few sensor nodes are physically compromised.

3.1.2.3 Two phase session based key management

Pietro et al., [130] proposed a key management protocol for large scale WSNs. The protocol is composed of two main phases. In the first phase, a new session key is generated, while in the second phase the new session key is distributed to all sensor nodes in the WSN. In the first phase, each sensor node autonomously generates the session key. The algorithm driving such a generation makes sure that each sensor node generates the same key. The second phase focuses on ensuring that each sensor node holds an

appropriate set of cryptographic keys. This second phase is needed for synchronization. Similarly like BROSK this scheme has not considered node capture attacks.

3.1.3 Hierarchical Based Key Management

3.1.3.1 LEAP

LEAP is based on the theory that different types of messages exchanged between nodes need to satisfy different security requirements. All the packets transferred in a sensor network need to always be authenticated where a sensor node knows the sender of the data since an adversary may attack a WSN with false data at any time. On the other hand confidentiality, like encryption of packets carrying routing information, is not always needed. Different keying mechanisms are necessary to handle the different types of packets. For this Zhu et al., [121] establish LEAP with four types of keys that must be stored in each sensor: individual, pair-wise, cluster, and group. Each key has its own significance while transferring messages from one node to another in a WSN. By using these keys LEAP offers efficiency and security with resistance to copious attacks such as the wormhole and Sybil attacks. LEAP uses μTESLA for local broadcast authentication.

The advantages of this scheme include that it offers efficient protocols for supporting four types of key schemes for different types of messages broadcast, it reduces battery usage and communication overhead through in-network processing, and it uses a variant of μTESLA to provide local broadcast authentication. The disadvantages of this scheme include that it requires excessive storage with each node storing four types of keys and a one-way key chain. In addition, the computation and communication overhead are dependent upon network density (the denser a network, the more overhead there is).

3.1.3.2 Cluster based key management for WSN

Jolly et al., [131] structure the WSN in clusters, and then assign one gateway (super node) to each cluster to be in charge of the cluster. Gateway nodes are equipped with more resources compared to the rest of the nodes. In their solution each sensor stores two keys. One key is shared with a gateway and the other with a sink. This scheme can also

be categorized under heterogeneous WSNs. The disadvantages of this scheme include that in case a gateway node is compromised, it means that the data confidentiality and communication of this cluster will be compromised..

3.1.3.3 Three tier key management for WSNs

Bohge et al., [132] propose a new WSN structure for their key management idea. They use a three-tier ad hoc network topology. At the top level there are high-power access points that route packets received via radio links to the wired infrastructure. On the second level there are medium power forwarding sensor nodes and at the bottom level there are low power mobile sensor nodes with limited resources. The lower level nodes share keys with the level above them. For more security each sensor node should have a personal initial certificate. They split sensing data into two parts: normal and sensitive data.

3.1.3.4 SHELL

Younis et al. [127] propose a lightweight combinatorial construction of key management for clustered WSNs, called *SHELL*. In SHELL, collusion is reduced by using nodes' physical locations for computing their keys. This scheme uses a command sensor node to govern the entire WSN. The command sensor node directly communicates with the gateway nodes which are in charge of individual clusters. Sensor nodes can be added to this WSN at any time. The gateway nodes are powerful enough to communicate with the command sensor node and undertake required key management functions.

Each gateway node can communicate with at least two other gateway nodes in the WSN, and has three types of key [127]. The first is a preloaded key that allows the gateway to directly communicate with the command node. The second type of key allows the various gateway nodes to communicate. The third allows the gateway to communicate with all the sensor nodes in its cluster. In the case of node capture attacks, it is assumed that the command node is unable to be compromised. If a single sensor in a cluster is compromised, the keys of all the sensor nodes in the cluster have to be replaced. If a

gateway node is compromised, the command node will do rekeying of the inter-gateway nodes. The application scenario is not clearly explained. The re-keying action in clusters or for gateway nodes in the case of a single node compromise is costly in terms of resources.

3.1.4 Key Management for Heterogeneous Sensor Networks

In the work [122, 123], Du et al. considered key management in a *Heterogeneous Sensor Network* (HSN) that consists of a small number of powerful high-end sensors and a large number of low-end sensors.

They have also presented an effective key management scheme – the asymmetric pre-distribution (AP) scheme for HSNs [122]. Powerful high-end sensors are utilized to provide simple, efficient and effective key set up schemes for low-end sensors. The basic idea of the AP key management scheme is to pre-load a large number of keys in each high-end sensor while only pre-loading a small number of keys in each low-end sensor. A high-end sensor has much larger storage space than a low-end sensor, and the keys pre-loaded in a high-end sensor are protected by tamper resistant hardware.

However according to Hussain et al. [145] the AP scheme is not efficient in terms of memory overhead. For example, if there are 1000 low end-sensors and 10 high end-sensors in an HSN and each high end-sensor is loaded with 500 keys and each low end-sensor is loaded with 20 keys, the total memory requirement for storing these keys will be $(10 \times 500) + (1000 \times 20) = 25{,}000$ (in the unit of key length).

Du et al. also propose a routing-driven key management scheme, which only establishes shared keys for neighbour sensors that communicate with each other [123]. They have used tree topology based routing; and also elliptic curve cryptography to further increase the efficiency of the key management scheme.

There are a few issues in their current solution. The authors have presented an evaluation comparison with homogenous WSNs, whereas the current proposed solution is for heterogeneous WSNs. The authors have not considered the effect of node capture attacks when high-end sensor nodes are compromised. This is important since all

communications between clusters is through the high-end sensor nodes. According to Hussain et al. [145], asymmetric cryptography such as RSA or elliptic curve cryptography (ECC) is unsuitable for most sensor architectures due to its high energy consumption and increased code storage requirements. As target applications for the scheme have not been clearly described in the paper, it's therefore difficult to establish whether the current network model can achieve scalability.

Hussain et al. [145] have also proposed a key distribution scheme for heterogeneous WSNs. Similarly they have assumed high-end (H-end) and low-end (L-end) sensor nodes. H-sensor nodes will act as cluster heads. There is a key pool K consisting of M different key chains. These key chains will be used to preload keys in L-end sensor and H-end sensor nodes. After a successful cluster formation phase a shared key discovery phase begins. Every L-end sensor will establish a key with a H-end sensor and with its own neighbouring nodes.

Similar to Du et al. in [122, 123], Traynor et al. [124] also assume that there are sensor nodes in the WSN that are more powerful and more secure than others, and these more powerful sensor nodes are also in tamper proof boxes or well guarded. A sensor node that has limited memory and processing power is identified as L1, and a sensor node that has more memory and more processing power is identified as L2 [124]. L2 nodes act as head sensor nodes for the L1 sensor nodes and have the responsibility of routing packets throughout the WSN. These L2 sensor nodes have access to gateway servers which are connected to a wired network.

3.1.5 Group Based Key Management

Most proposed solutions for group key management use a session key concept. Here we will provide an overview of some of these solutions.

Eltoweissy et al. [77] propose a scheme for group key management in large-scale WSNs. Their proposed scheme is based on *Exclusion Basic Systems* (EBS). The use of EBS (n, k, m) is to assign and manage keys for a group, where n is the number of group members, k represents keys held by the nodes and m is the number of broadcast messages needed for

rekeying after a node is evicted. They assume that all sensor nodes are pre-initialized before deployment, with an identical state mainly consisting of a set of training parameters and a number of keys. A key server also has one or more session keys known to subsets of group members. All group members aware of a particular session key constitute a secure communication group. Members in a secure communication group use the session key corresponding to the group for the encryption of messages exchanged among group members.

Pietro et al., [130] propose a key management solution for large scale WSN. Their protocol generates keys without requiring communication among sensors. They believe that direct communication between sensor nodes consumes more energy. They also prefer to use the session key concept. They propose two different methods for sensor nodes to agree on session keys: one for a base station scenario and the other for a completely distributed scenario. The base station has to interact with the WSN to invoke the command to generate new keys. In the distributed case each sensor node stores a parameter μ that drives the generation of a new session. After a time out of μ clock ticks has elapsed, the sensor node invokes the generation of a new session key.

Group communication applications can use IP multicast to transmit data to all n group members using minimum resources. Efficiency is achieved because data packets need to be transmitted only once when they pass through any link between two nodes, hence saving bandwidth. This contrasts with unicast-based group communication where the sender has to transmit n copies of the same packet. Any multicast-enabled host can send messages to its neighbour router and request to join a multicast group [39]. There is no authentication or access control enforced in this operation [145]. The security challenge for multicast is in providing an effective method for controlling access to the group and its information that is as efficient as the underlying multicast.

After explanation of all these key management solutions, we have concluded that the main objective of these solutions is the same: secure communication between pairs of sensor nodes or sources and destinations. However all these solutions are applications or structure dependent and limited to specific applications. These static WSN security

solutions do not support mobility, which results in significant limitations. Mobility generates more security challenges and attacks than in static WSNs.

Furthermore these solutions only describe resilience against node capture attacks but fail to discuss the possible attacks that can occur after node capture, e.g. replication, black hole and Sybil attacks. Key management also has some inter-link issues related to network processing (such as data aggregation). Suppose that an aggregator, a cluster head, a master sensor node or any ordinary sensor node is compromised where data aggregation takes place. This would bring issues surrounding data confidentiality, data integrity and trust to the fore.

In case a group leader, a cluster head or an aggregator sensor node becomes compromised, there should be a solution to allow election or selection of a new group leader, a cluster head or an aggregator sensor node, in order to provide better and continuous service and availability.

All these issues (mobility, secure data aggregation, secure group leader election or selection and resilience against all other possible attacks) are related to key management directly, so we will present a brief literature related to these issues next.

3.2 Secure Data Aggregation

In the initial stages of sensor network research, many data aggregation protocols [58-59, 147] were proposed, but none of them were designed with the consideration of possible security threats. Further research in this area highlighted the importance of security.

Hu et al., [150] proposed a secure hop-by-hop data aggregation scheme. In this scheme individual packets are aggregated in such a way that a sink can detect non-authorized inputs. The proposed solution introduces a significant bandwidth overhead per packet. They also assume that only leaf nodes with a tree-like network topology sense data, whereas the intermediate nodes do not have their own data readings. Jadia et al., [148] extended the Hu et al. approach by integrating privacy, but considered only a single malicious node.

Several secure aggregation algorithms have also been proposed for the scenario of a single data aggregator for a group of sensor nodes. Przydatek et al. [149] proposed *Secure Information Aggregation* (SIA) to detect forged aggregation values from all sensor nodes in a network. The aggregator then computes an aggregation result over the raw data together with a commitment to the data based on a Merkle-hash tree and then sends them to a trustable remote user, who later challenges the aggregator to verify the aggregate. They assume that the bandwidth between a remote user and an aggregator is a bottleneck. Therefore their protocol is intended for reducing this bandwidth overhead while providing a means to detect with a high probability if the aggregator is compromised. Yang et al. [152] describe a probabilistic aggregation algorithm which subdivides an aggregation tree into sub trees, each of which reports its aggregates directly to the sink. Outliers among the sub trees are then probed for inconsistencies [56]. Moreover a number of aggregation algorithms have been proposed to ensure the data confidentiality of the data against intermediate aggregators. Such algorithms have been proposed by Girao et al. [151], Castelluccia et al. [61], and Cam et al. [147]. In the next section we will describe the relationship between data confidentiality and secure data aggregation.

3.2.1 Confidentiality and Data Aggregation

Confidentiality can be maintained between sources and destinations using different solutions according to requirements and available resources. In our further discussion we describe three possible solutions to achieve end-to-end (source to destination) confidentiality in large scale sensor networks.

The first option is for each sensor to store a unique key shared only with the sink, and to send encrypted data through other sensors to the sink without decryption at any non-sink node. This end-to-end confidentiality can be achieved but there are some drawbacks. Because all packets are forwarded towards the sink, a lot of bandwidth is consumed. It's also very burdensome when a sensor network is busy to recover large amounts of data from every single node in the case of data loss. Finally, there is an extreme imbalance

between sensors in terms of the amount of data communicated. For example sensors closer to the sink will lose energy more quickly.

The second option is hop-by-hop secure data aggregation. This type of scheme is only limited to specific topologies such as tree topologies, but it nonetheless achieves the goal.

The third option which we recommend is where an aggregator node has an appropriate position, such as a group leader node, which can assist in securely aggregating the data within its group. This will also help to reduce the amount of data to be transferred to the sink and support network structure independence. It is possible that the group leader node is not in the best position to perform the role. The issue of how to optimally select a node in a group for the role of data aggregation in relation to several factors such as minimal energy consumptions and the number of neighbouring sensor nodes will be presented later in this thesis.

3.2.2 Homomorphic Encryption

Homomorphic encryption is a semantically-secure encryption which, in addition to providing the standard guarantees has additional properties. In particular the sum of any two encrypted values is equal to the encrypted sum of the values. There are several efficient homomorphic cryptosystems such as *Unpadded RSA*, *El-Gamal*, *Goldwasser-Micali* and *Benaloh and Paillier* [153].

Consider Unpadded RSA as an example, where we use the notation $E_k(x)$ to denote the encryption of a message x with a key k using the method. Suppose that a public RSA key is expressed as $pk = (e, m)$. Then the encryption of a message x with key pk is signified as $E_{pk}(x) = x^e \bmod m$. In this case we have the homomorphic property:

$$\begin{aligned}(E_{pk}(x_1)\, E_{pk}(x_2)) \bmod m &= (x_1{}^e x_2{}^e) \bmod m \\ &= (x_1 x_2)^e \bmod m \\ &= E_{pk}(x_1 x_2)\end{aligned}$$

For addition and average calculations in WSNs, we can use the following homomorphic encryption algorithm.

Encryption:

1. Represent a message (sensed data) as an integer d with $0 < d < Z$, where Z is a large integer.
2. Let $s = (k, Z)$ be a shared secret key with $0 < k < Z$.
3. Define $c = E_s(d) = (d + k) \bmod Z$.

Decryption:

1. Compute $E_s^{-1}(c) = (c - k) \bmod Z = d$.

Addition of Ciphertexts:

1. Let $s_1 = (k_1, Z)$ and $s_2 = (k_2, Z)$ be two secret keys, and d_1 and d_2, where $0 < (d_1 + d_2) < Z$, be two messages. Compute $c_1 = E_{s1}(d_1) = (d_1 + k_1) \bmod Z$ and $c_2 = E_{s2}(d_2) = (d_2 + k_2) \bmod Z$. This leads to $(c_1 + c_2) \bmod Z = ((d_1 + d_2) + (k_1 + k_2)) \bmod Z = E_{s1,2}(d_1 + d_2)$, with key $s_{1,2} = (k, Z)$ and $k = (k_1 + k_2) \bmod Z$.
2. For decryption, we have:

   $E_{s1,2}^{-1}(c_1 + c_2) = ((c_1 + c_2) - k) \bmod Z = d_1 + d_2$.

Note that if n different ciphers c_i are added, then Z must be larger than $\sum_{i=1}^{n} d_i$, and otherwise the correctness of recovered data is not provided. In practice, if $p = \max(d_i)$, then Z should be selected as $Z = 2^{[\log 2 (p * n)]}$ [61].

3.3 Replication Attacks

Replication attacks can be launched in two different ways in WSNs. In the first type of replication attack an adversary can eavesdrop on communications and resend old packets again multiple times in order to waste its neighbouring sensor nodes' energy. In the

second type of replication attack an adversary can insert additional replicated hostile sensor nodes into the WSN after obtaining some secret information from captured sensor nodes or through infiltration [133, 134].

Fu et al. evaluated the effect of replication attack on key pool based key management schemes [72, 75, 128, 135, 136]. They analyze, characterize and discuss the relationship among the replicated hostile sensor nodes, the WSN, and the resilience of various random key pre-distribution schemes against replication attacks using a combination of modelling, analysis, and experiments. Example findings include the following.

(1) WSNs with random key pre-distribution schemes, even with one replicated sensor node, start to become almost 100% insecure when an adversary captures and stores key information equivalent to those carried by one good sensor node.

(2) When the replicated sensor node has less memory to store key information than the original sensor node, among the proposed schemes, the q-composite scheme with larger q is most resilient against replication attacks while the Basic Scheme is least resilient [133, 134]. In Parno et al. presented distributed methods of detecting replication attacks in WSNs [137]. According to Parno et al. an adversary can compromise a few sensor nodes in the network and can create more cloned sensor nodes to place them in different locations in the WSN to launch replications attacks. We conclude from our review of the literature that group based key management schemes are more resilient against replication attacks, even after some of the nodes have been compromised. This is due to the fact that group based key management schemes can minimize global key sharing in comparison to key pre-distribution schemes [137].

3.4 Secure Group Leader Election/Selection

There are various different proposed solutions for group leader election or selection; some of these also consider security issues. First we will give an overview of non secure group leader or cluster head election related work, and then we will briefly describe related work on secure group leader election/selection.

In *LEACH* [139], initially when clusters are being created, each sensor node decides whether or not to become a cluster head for the current round. This decision is based on the suggested percentage of cluster heads for the WSN (determined *a priori*) and the number of times the sensor node has been a cluster head so far.

Wen et al [140] define two different methods: centralized and distributed for cluster head election. In the centralized method, the current cluster head, sensor *i*, determines a new cluster head by aggregating energy and neighbour sensor nodes information from its cluster members. In the distributed method, once the energy in the current cluster head is below a given threshold, it transmits a message to start the reselection process. Each cluster member then checks its energy constraints. As long as the cluster member satisfies these constraints, it generates a random waiting time which depends on the number of neighbouring cluster members and the remaining energy level.

Vasudevan et al. [146] define algorithms for the secure election of leaders in wireless ad-hoc networks, but these algorithms use public-key cryptography, which is unappealing for resource-constrained WSNs. As described earlier, security in WSNs is subject to different and increased constraints compared to traditional and ad-hoc networks. Keeping the limited resources of WSNs in mind, Liu et al., [128] present a hybrid public-key system for WSN security, where more capable gateway devices perform the bulk of the computation. This approach is appealing in group mobility situations, but again it may be limited to certain applications as it is still not as computationally lightweight as a symmetric-key algorithm would be. Symmetric-key authentication systems are a well-researched field. In order to use symmetric-key cryptography in a WSN, we must have a reliable method for the distribution of symmetric keys. There has been work in this area. As described in section 3.1.3.1, LEAP is a method for key management and authentication in WSNs. However, LEAP doesn't consider group leader election in the case of group leader compromise. As described in section 3.1.2.1, Perrig et al. present a number of security protocols for WSNs using a master key system, where each sensor node derives its private key from the master key. Furthermore as we have discussed in section 3.1.1.1, Eschenauer and Gligor present a probabilistic keying scheme that relies on sensor nodes being assured a certain probability of communication with other sensor nodes, rather than being assured with certainty, to reduce the number of keys a node must

keep track of Traynor et al. [138] present LIGER, a hybrid system for symmetric-key distribution using both probabilistic keying and a KDC in MSNs, and an implementation and analysis of probabilistic keying schemes [124]. Using elements from LIGER and Traynor et al, we can ensure that sensor nodes have a computationally inexpensive means of communicating securely, and in turn, electing a leader securely.

3.5 Key Management for MSNs

Security issues can be more destructive in MSNs then static WSNs. For example, in key pool based schemes [72, 75, 78], if a single mobile sensor node is compromised it can listen to the communication of the entire WSN due to the global sharing of keys. Currently key management has only been considered in mobile ad hoc networks and most of these proposed solutions consider either hierarchal key management or group based key management.

Wang et al. [143] propose a hierarchical key management scheme for secure group communication in mobile ad hoc networks. In this scheme the entire network is split in groups and further into sub-groups. Subgroups are further divided into two levels, L1-subgroups and L2-subgroups. Different keys are used at each of these levels. For communication between groups, a communication key is used by bridge nodes referred to as communication nodes. There are certain issues when a new node joins a group. In particular, the L2-head has to regenerate a new subgroup key and send it to its entire set of members.

Wu et al. [141] propose a secure and efficient key management scheme for mobile ad hoc networks. They also organise their network into server groups and use public key infrastructure. Along with key management they also explain about group maintenance and formation. Each server group creates a view of the certificate authority (CA) and provides a certificate update service for all nodes, including the servers themselves. A ticket scheme is introduced for efficient certificate service provision. As they use an asymmetric cryptographic method, it is not efficient to use such a method for WSNs.

Cho et al. [142] propose a region-based group key management scheme for mobile ad-hoc networks. In this scheme group members are broken into region-based subgroups. The leaders in a subgroup securely communicate with each other to agree on a group key in response to membership changes and member mobility-induced events. They have assumed that every single node is equipped with GPS and knows their location when they move across regions. Such an assumption is not suited for WSNs due to its limitation. They have used a hierarchy of keys, e.g. regional keys, group keys and leader keys at different levels of the group. In their attack model they assume only external attacks can occur. However in WSNs node capture attack is totally different and unique to traditional network schemes. If we apply node capture attacks at different levels of a group on [142], or if a Leader node becomes compromised, data confidentiality and integrity can be compromised and there is no election or selection method for Leader nodes.

We can see from existing key management solutions for mobile ad hoc networks that most of these solutions are hierarchical. There are some issues that relate to application of these solutions for WSNs. WSNs are scaleable networks, public key management and GPS are not ideal for WSNs due to the limited resources of nodes and regeneration of keys due to the joining and leaving of nodes in a group is an energy consuming task, especially when we consider sensor networks in water (such as the sea) where nodes may change position rapidly. Therefore it remains an open research issue, and many solid answers are needed in order to provide for future solutions.

3.6 Summary

In this chapter, we surveyed the literature and works relating to key management, secure data aggregation, group leader election/selection and key management for MSNs. We presented and discussed the existing solutions of key management for static WSNs. These solutions were classified into five different types including: key pool based key management, session based key management, hierarchical based key management, key management for heterogeneous sensor networks, and group based key management. All of these solutions place emphasis on the important issue of providing high resilience against node capture attacks and providing better secure communication between sources

and destinations. This chapter pointed out the main drawbacks of existing key management solutions. There are some common drawbacks in these schemes, for example all of these solutions are structure dependent and any change in the structure directly affects the security of the WSN. Each key management solution is particularly designed for one specific problem, and these solutions do not handle problems such as secure data aggregation or replication attacks. Furthermore we have listed some individual weaknesses of these schemes are summaries in table below:

Key Management Scheme	Drawbacks
Key pool based	• Probabilistic key sharing increases the probability of compromising the entire WSN. • Extra communication overhead during key establishment phase. • Memory overhead due to preloading more keys to increase the probability of key sharing between sensor nodes. • Forward and backward secrecy requirements will increase communication overhead.
Hierarchical	• Minimum number of routes for child sensor nodes to communicate with their parent sensor nodes. Therefore in the case of a parent sensor node being compromised, all of the communication from its child sensor nodes will be blocked. • These key management schemes can only be used in specific applications and they are not scalable.
Session based	• These solutions are limited to small WSNs. • Sharing common values to produce different session keys for encryption increases vulnerability of communication compromise of the WSN in the case of compromise of a single sensor node. • These solutions are more vulnerable to DoS attacks.
Heterogeneous	• Physical compromise of a high end-sensor causes problems similar to the compromise of a parent sensor node in the hierarchical key management. • Some of these schemes have used asymmetric cryptography which is unsuitable for most sensor architectures due to its high energy consumption and increased code storage requirements.

	• Forward and backward secrecy is another weakness of these schemes.
Group based	• These solutions only describe resilience against node capture attacks but fail to discuss the possible attacks that can occur after node capture, e.g. replication, black hole and Sybil attacks. • In case of aggregator, a cluster head, a master sensor node is compromised where data aggregation takes place. This would bring issues surrounding data confidentiality and integrity. • There is no alternative in case a group leader, a cluster head or an aggregator sensor node becomes compromised, in order to provide better and continuous service and availability.

Table 3-1: Drawbacks of current key management schemes

Later in the chapter we have described related works concerning secure data aggregation and its importance. Secure data aggregation is vital in applications where very sensitive data are communicated through the sensor nodes. Compromise of an aggregator node can be a significant risk to data confidentiality. Therefore data aggregation should be performed on encrypted data without decryption to improve data confidentiality. Furthermore we have presented related works about secure group leader selection. Finally we described related work about key management for MSNs.

Chapter Four: Structure And Density Independent Group-Based Key Management (SADI-GKM) Protocol Design Overview

The unique properties of WSNs increase their popularity and potential for future involvement in major applications of our daily life. All of these facilities are potentially hindered by a number of important issues. It is therefore important that these issues be resolved in order to fully benefit from WSNs. Some of these issues are general and basic, e.g. scalability and resource usage reduction, and have influence or inter-relations with other issues such as security, routing and fault tolerance. For example, security and routing protocols must also be scalable and energy efficient.

In this chapter we will concentrate on the issue of security, presenting the requirements and design overview of a key management protocol that is intended to improve security while at the same time taking into account the more general issues of scalability and energy efficiency as described above. Having described our requirement and protocol framework in this and detail design the next chapter, we will then go on to test it and evaluate it against other protocols in Chapter six.

4.1 Background

As described in earlier chapters, WSNs can contain hundreds and potentially thousands of small sensor nodes able to perform various different jobs. It is commonly assumed that the purpose of WSNs is to monitor large areas. Moreover the number of WSN applications is increasing quickly due to their unique characteristics. Future applications will be highly scalable, e.g. whole countries and cities will be monitored for various purposes using WSNs. Therefore scalability is a core issue and this can affect the

performance of any proposed security and non-security protocols especially when we also take into account resource limitations. To handle scalability issues using fewer resources, researchers spilt an entire network into zones, groups or clusters [31]. Furthermore, these groups are then organised into different topologies according to the application requirements. Clustering and group management are well known approaches used to provide efficient management of large scale networks of various types, including WSNs and ad hoc networks [31].

Our aim is to provide secure communication for scalable WSNs. We have therefore used group key management concepts in our protocol design.

Furthermore in relation to security and scalability, a WSN is strictly structure and application dependent, i.e. if a protocol is designed for indoor applications it is unlikely to work in outdoor applications. If the size of a network increases or decreases these protocols often show appalling performance, e.g., μTESLA [39] shows good performance in small scale WSNs but its performance decreases in large scale WSNs [20]. Furthermore μTESLA is also topology dependent and especially designed for hierarchical WSNs.

As described in section 2.3.1 about node capture attack, the sensor nodes and their operation in remote and hostile areas makes it easy for sensor nodes to be captured and increases the chance that a sensor node will be targeted. There can be many possible ways for an adversary to use a compromised sensor node. For example, after compromising a sensor node or nodes physically, an adversary can easily extract key information from the sensor node and replace it back into the WSN as an adversary-controlled sensor node. It will then behave as a normal sensor node and can establish communication with other non-compromised sensor nodes. Consequently data confidentiality and privacy will be at high risk. Therefore any solution to provide resilience against node capture should assure that the compromised sensor node should not leak sensitive data and should not compromise the confidentiality of other sensor nodes.

To minimize the number of transmissions from thousands of sensor nodes towards a sink, a well known approach is to use in-network aggregation as discussed in section 2.2.10

and 3.2. A serious issue connected with in-network data aggregation is data security/integrity [154]. There are different types of attacks which can be harmful for in-network data aggregation, e.g. a compromise of an aggregator sensor node due to physical tampering using node capture attack. In many applications nodes are communicating highly sensitive data (for example in military and rescue applications) and due to such threats data integrity/security is vital. Therefore a secure data aggregation method is required to provide better integrity and data confidentiality.

Furthermore, in case of an aggregator, a group leader or a cluster head sensor node failure due to resource depletion; there should be a secure way of electing or selecting a new aggregator or group leader sensor node. There are various proposed schemes which provide election of a new group leader sensor node, but they generally fail to consider the issue of security. For example, a new elected group leader sensor node could turn out to be an adversary sensor node. We therefore need to make sure that any new group leader sensor node is a trusted node.

Considering our previous discussion, it's clear that all of these issues – scalability, application or structure dependency, node capture attack, replay attack, secure data aggregation and secure group leader election – are inter-related with each other. By ignoring any one of these we are likely to end up weakening the overall security solution.

Until now all of these problems have been addressed individually but separately, and while any good solution for a single problem might tackle this problem well, it will often have no integration with other solutions proposed for the other problems. In this case the problem will remain with the overall method. Therefore an integrated approach is required to combine all of these issues together and identify their relationships, allowing them to be handled efficiently.

Following from our research into node capture attacks, various evolutionary results have highlighted three main factors which can aid adversaries during node capture attacks, in order to compromise the communication of the entire sensor network [33, 71]. The first factor is that node capture attacks can be a threat only if sensor nodes within the WSN share a key or keys with neighbouring sensor nodes used to encrypt or decrypt data. Consequently the greater the level of key-sharing between neighbouring sensor nodes, the

greater the threat to data communication integrity and confidentiality. Most existing key management solutions suffer from this drawback. The second important factor is the structure (topology) of the WSN. In general, the fewer the communication links between sensor nodes, the greater the possibility that an attacker can entirely block the communication paths between a source and destination. For example, node capture attacks are generally more effective in tree topologies than mesh topologies because in the former there is only one route from child to parent. If the parent node is compromised, the entire communication from its child nodes downward will potentially be compromised [33, 71]. The last factor which has a direct influence on node capture attacks is the density of the network, having a similar effect as the second factor. For example, physical compromise of sensor node with high density will be larger damage as compare to sensor node with low density. Therefore these three design factors were ascertained as being vital for a key management protocol to provide high resilience against node capture attacks. First, encryption should happen only at source sensor nodes and data should be decrypted at the corresponding destination sensor nodes (group leaders or sinks) to provide better confidentiality. The second factor is the selection of an appropriate topology, since this can help to provide resilience against node capture attacks. For example, DGKE has better performance results against attacks when a random mesh is used as compared to other topologies. Finally we need to select a suitable density according to the application in order to overcome the risk of serious damage to the WSN resulting from a node capture attack. In the next chapter we will describe the details and evaluation of DGKE.

4.2 Aims and Objective

In this section, we discuss the overall aims and specific objective of our work. In previous sections we have described relevant security threats and issues which are inter-linked with each other, and the need for an integrated security solution. We also provided background needed in order to support our main aims and objectives. Given this, the major research objectives that we address in the problem area are as follows:

- To develop a novel key management protocol. This protocol should have different layers to tackle different security problems at the prevention level. All these layers must be integrated with each other to prevent an attacker from system penetration. The proposed key management protocol should support topology independency, scalability, node to node authentication, high resilience against node capture attacks and replay attacks, and be resource efficient. The protocol should be evaluated using different topologies and compared against existing key management protocols to show its performance. This can be carried out using simulations.
- To design a secure data aggregation scheme to provide better data confidentiality and integrity inside groups. This secure data aggregation service will come under the second layer of the key management protocol, and should not increase resource usage. The second layer should provide different options in terms of data confidentiality according to user and application requirements. This mechanism can be evaluated using simulations by monitoring the aggregator sensor node energy consumption.
- To develope novel techniques for secure selection of a new group leader sensor node using various selection parameters. The proposed techniques should avoid the selection of sensor nodes with low trust, low energy or which have long distances from the current group leader. The selection process should be quick and short to minimize energy consumption. The proposed schemes can be evaluated by comparing them to existing solutions using simulations. This scheme will come under the third layer of the proposed protocol.
- To design a key management scheme to support MSNs appropriately. The proposed scheme should support different mobility patterns (e.g. free and guided mobility) and different security levels according to different applications and resource availability. This scheme can also be evaluated using simulations.

4.3 Requirements

I. *Confidentially and Data integrity:* In WSNs, confidentiality relates to the following:

- Providing confidentiality of the wireless communications channels to prevent eavesdropping. A sensor node should not leak sensor readings to its neighbouring sensor nodes. In some applications, the data stored in the sensor node may be highly sensitive. To achieve this, a sensor node should avoid sharing keys with neighbouring sensor nodes, when these keys are used for encryption and decryption [33].
- The building of secure channels in the WSN.
- Sensor information, such as sensor identities and public keys, is encrypted to some extent to protect against traffic analysis attacks.

  Data integrity issues in wireless networks are similar to those in wired networks. Data integrity ensures that any received data has not been altered or deleted in transit. We should keep in mind that an adversary can launch modification attacks when cryptographic checking mechanisms such as message authentication codes and hashes are not used.

II. *Authentication:* In WSNs an adversary's attack is not just limited to modifying data packets. An adversary can change the whole packet stream by injecting additional packets, so the receiver node needs to ensure that the data used in any decision-making process originates from the correct source [41]. Authentication of other sensor nodes, cluster heads, and sinks before revealing information is therefore crucial.

III. *Availability:* Key management services must ensure that confidentially and group-level authentication services are available to authorized parties when needed. However strict limitations and unnecessary overheads weaken the availability of sensors and WSNs [41].

IV. *Scalability:* Key distribution schemes must support large scale WSNs, and must be flexible during increases in the size of the WSN.

V. *Data freshness:* Proposed key management solution should have the ability to check data freshness that the data is recent, and it ensures that no old messages have been replayed.

VI. *Application independency:* Due to disorder and the wide range of potential applications for WSNs, an application independent solution is required which can be used for more than one application. This is especially important in MSNs and ad hoc sensor networks where topologies change frequently due to nodes joining and leaving. A topology independent solution could significantly decrease energy consumption.

VII. *Survivability:* Due to the unattended nature of WSNs, an attacker could launch various security attacks and even compromise sensor nodes without being detected. Therefore, a WSN should be robust against security attacks such as node capture attacks and replication attacks. Even if an attacker succeeds, its impact should be minimized. For example, the compromise of a single sensor node should not break the security of the entire WSN.

VIII. *Supporting secure in-network processing:* Security mechanisms should permit in-network processing operations in a secure way such as through secure data aggregation. In-network processing significantly reduces energy consumption in WSNs.

IX. *Supporting secure group leader selection:* In case a group leader sensor node is compromised or dies, there should be a secure way of selecting another group leader sensor node.

X. *Forward and backward secrecy:* The proposed key management scheme should provide forward and backward secrecy when nodes join or leave a group [178].

XI. *Memory overhead:* Propose key management should use few keys while supporting a high level of security.

XII. *Connectivity:* With a smaller number of keys, the probability that two sensors sharing at least one common key during any given time-interval should be kept as high as possible, in order to increase connectivity.

4.4 The Importance of Integrated Security

According to Yang et al. [156] there are basically two approaches toward security solutions in ad hoc networks (such as WSNs): proactive and reactive. The proactive approach tries to prevent an attacker from launching attacks in the first place, typically through various cryptographic techniques (such as key management). In contrast, the reactive approach seeks to detect security threats afterward and react accordingly. Due to the absence of a clear line of defence, a complete security solution for WSNs should integrate both approaches and include all three components: prevention, detection, and reaction.

The prevention component prevents an attacker from penetrating the system. However, the history of security has clearly shown that a completely intrusion-free system is infeasible, no matter how carefully the prevention mechanisms are designed [156]. This is especially true in WSNs, since they consist of small sensor devices that are prone to compromise or physical capture. Therefore, the detection and reaction components that discover the occasional intrusions and take reactions to avoid persistent adverse effects are vital for security solutions to operate in the presence of limited intrusions. As argued by Nikopolitidis [3], security is a chain and is only as secure as the weakest link. Missing a single component may significantly degrade the strength of the overall security solution. Furthermore there are many threats at the prevention level. All these solutions aim to prevent adversaries from penetrating the system, and should be tightly integrated. However our proposed protocol falls into the category of a *prevention component* as shown in Figure 4-1.

4.5 Protocol Design Overview

We have designed our Structure And Density Independent Group-Based Key Management Protocol (SADI-GKM) [117] as a stack of four different layers with different functionalities, which are integrated with each other as shown in Figure 4-1. The protocol includes: a novel group-based key management scheme, efficient secure data aggregation [62], a novel secure group leader selection scheme and key management

capabilities for MSNs [36]. We have described earlier the need for such a protocol, in order to address the possible threats and issues together. Our proposed protocol has flexibility allowing the addition of more security solutions through the addition of more layers.

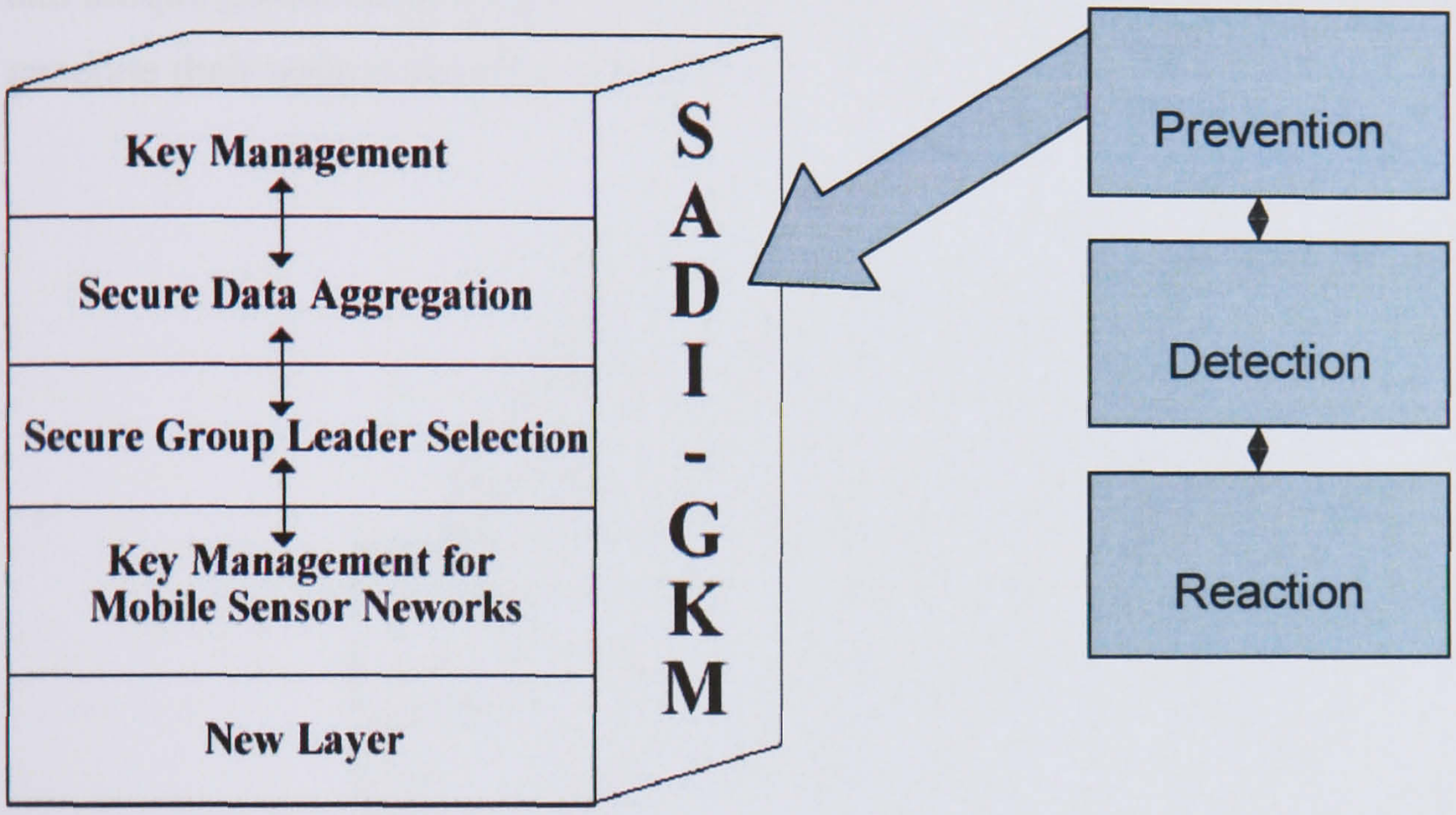


Figure 4-1: Structure and Density Independent Group-Based Key Management (SADI-GKM) protocol stack.

Our protocol works on three different node types: a sink node, group leader nodes and group member nodes. These three different node types play different roles in our protocol design and we have therefore designed different algorithms for each of them. The job of the group member sensor nodes is to sense, encrypt and send their data toward a sink or group leader sensor nodes. The group leader sensor nodes have multiple responsibilities as compared to the normal group member sensor nodes. These group leader sensor nodes will play the role of aggregators and gateways. The sink works as a base station to collect all the information and data from the sensor field. In the following subsections we describe the functionalities and tasks of each layer.

4.5.1 Layer 1: Key Management

This layer has responsibility to pre-establish keys between sensor nodes and provides basic rules and regulations which are further integrated with the other layers. We organise WSNs into multiple geographical groups as shown below in Figures 4-2 and 4-3. Every group of sensor nodes will be preloaded with a unique master key, authentication value and unique global network ID. All sensor nodes in every group will use this master key to generate their unique keys for encryption.

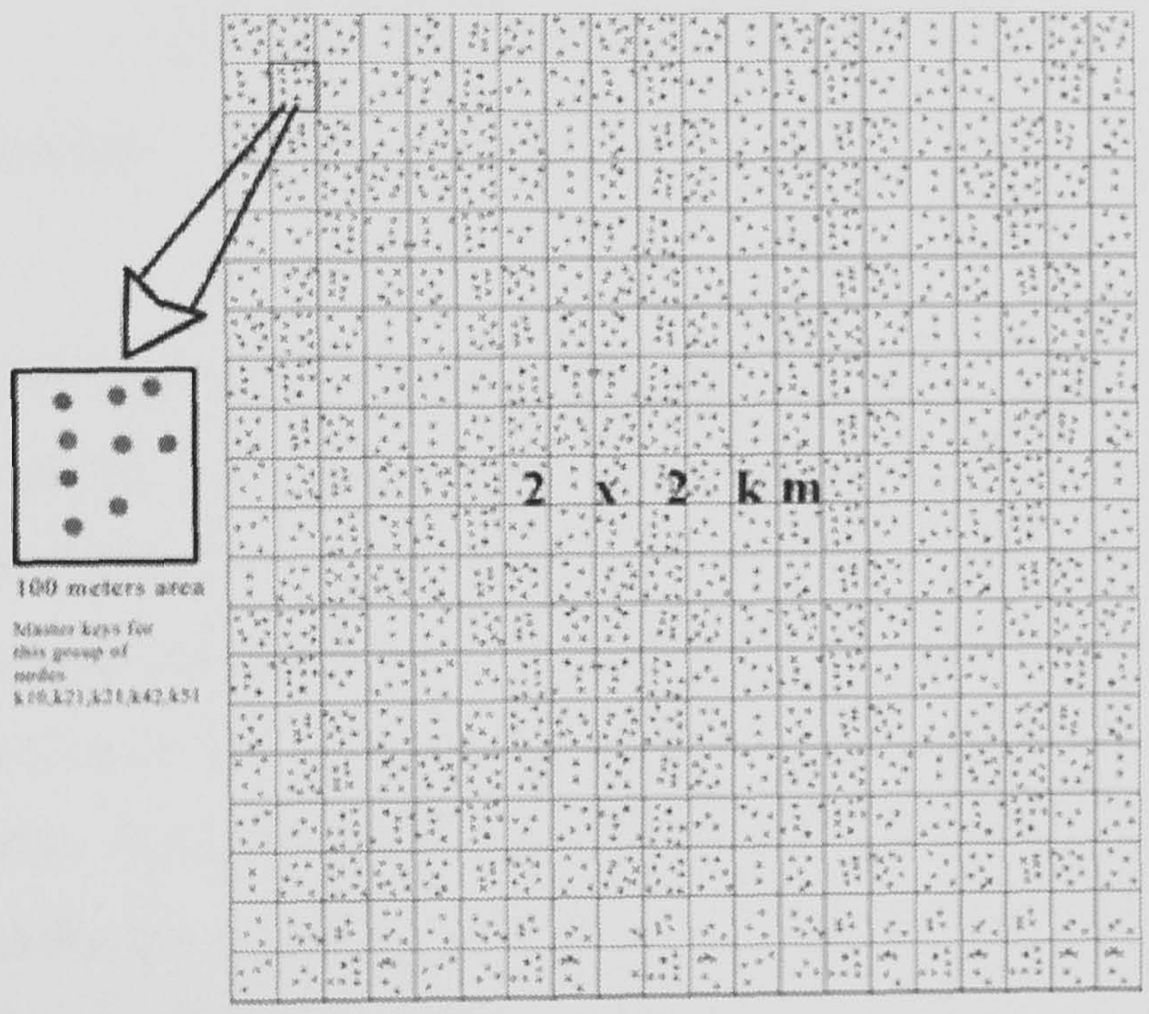


Figure 4-2: A 2 × 2 km outdoor WSN.

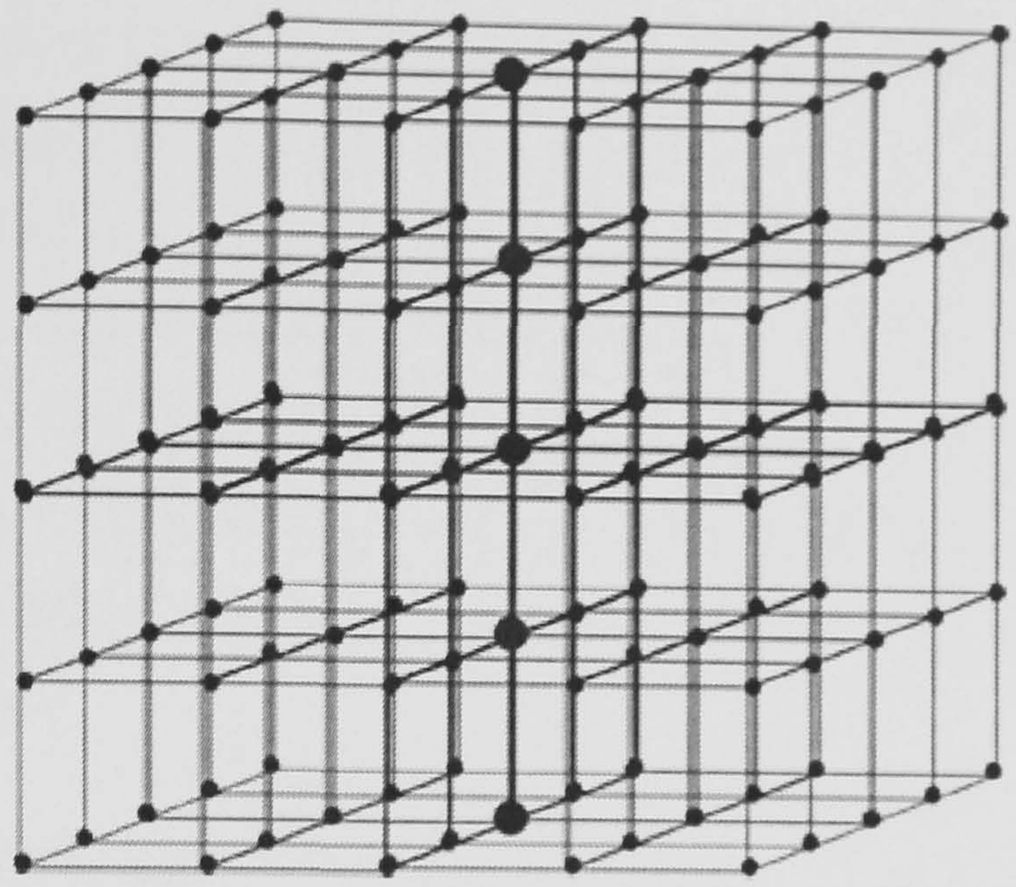


Figure 4-3: A WSN in a five story building (indoor WSNs).

The key management layer operates in two phases: a key pre-establishment phase and a data transmission phase. We try to avoid any communication between sensor nodes during the key establishment phase to reduce the risk of eavesdropping and store only a few keys in each sensor node in contrast to existing schemes [72-78][125, 126, 128, 130, 145]. which require every sensor node to be equipped with 50 to 100 keys and perform more communication during the key establishment phase. We believe that using our minimal pre-establishment approach will save considerable amounts of communication overhead and subsequently reduce the energy cost. The data transmission phase will begin after successful key establishment.

4.5.2 Layer 2: Secure Data Aggregation

As described in the previous chapter, aggregator sensor nodes receive data from member sensor nodes and calculate aggregated results to reduce the quantity of transmissions. In case of physical compromise of an aggregator sensor node, the data confidentiality and integrity of all other sensor nodes using the aggregator may also be compromised. We therefore propose two different cases for secure data aggregation. In the first case an aggregator sensor node (a group leader) authenticates incoming data, decrypts and aggregates it. Furthermore the aggregated result will be re-encrypted and sent toward the

sink. The sink will decrypt this incoming data in order to obtain the aggregated result. In the second case the aggregator sensor node will not be allowed to decrypt the data; aggregation will be performed on encrypted data. Moreover the sink will perform further calculations in order to obtain the aggregated results. In the second case we use homomorphic encryption. The secure data aggregation algorithms will be preloaded into all group leaders, group member sensor nodes and the sink. We have integrated this scheme with the first layer of our protocol. The implementation of second layer is described in section 6.3.3.

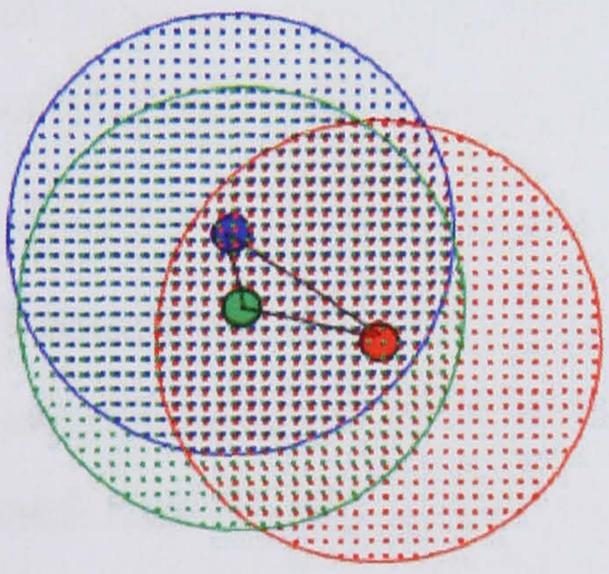
4.5.3 Layer 3: Secure Group Leader Selection

The current solutions for group leader election have only used energy as a major election criterion. In our proposed solution we consider four different criteria: available energy of a sensor node, the number of neighbouring sensor nodes, the communication distance from the current group leader node (based on the position of the new group leader node), and the trust level of a sensor node. We have assumed different values for the trust factors during our analysis. However, trust values can also be found in various other proposed solutions [157-159]. These factors are very important, for example, sensor nodes with low levels of trust should be avoided; and sensor node with fewer neighbouring sensor nodes should also be avoided. The position of the group leader is also very important as it can directly effect the energy consumption of the entire group (we explain this issue in detail in the next chapter). Similarly node energy plays an important role.

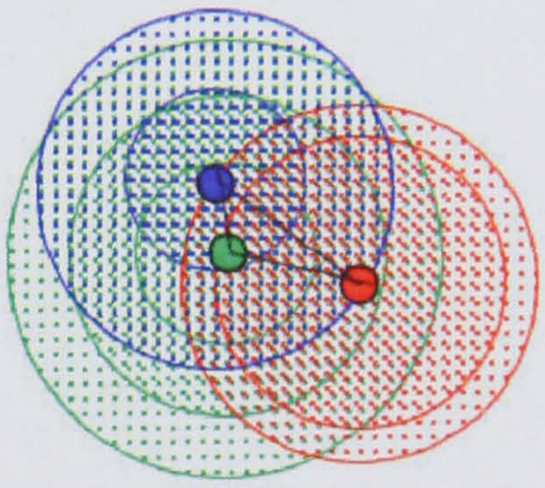
One of the main advantages of our scheme is that we do not involve all sensor nodes in the group for selecting a new group leader sensor node. First only neighbouring sensor nodes of the old group leader will be checked for new group leader selection.

Secure group leader selection/election is vital in case of a group leader failure due to a node capture attack, energy failure or other causes. Our proposed scheme can also be used for different applications with respect to different communication behaviour of the sensor nodes. For example, in some applications sensors might use different radio ranges with respect to distances between different sensor nodes. In this case we can measure the space between a source node and a group leader in order to establish a distance metric.

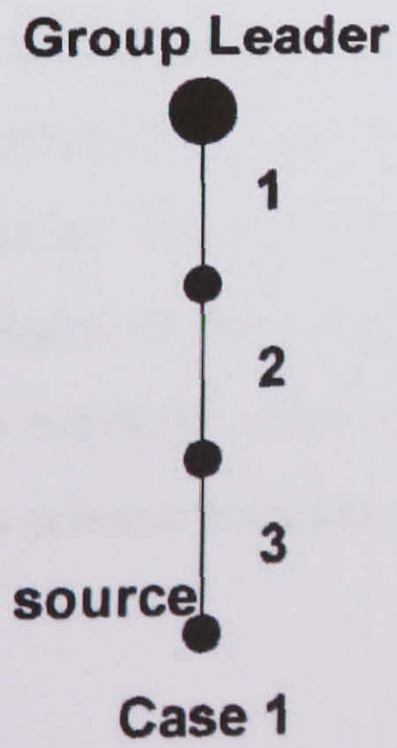
However in applications where a fixed radio range is used for communication between sensor nodes this may not be possible. In such cases we measure the space between a source sensor node and a group leader using a hop count. Figure 4-5 shows an example of this. In case 1 we have a fixed radio range meaning that every hop is considered to be the same distance, whereas in case 2 every sensor node has a different radio range, and different hops between sensor nodes can have different distances. The distinction is highlighted by the fact that in the diagram *d1* has four hops, *d2* has three hops, but the distance *d2* is greater then *d1*. Therefore in our proposed group leader selection process we have considered both of these cases. Furthermore our proposed secure group leader selection scheme can also be used in heterogeneous WSN applications where different types of sensor nodes are used. In the next chapter we will describe this process in detail using our proposed formula for group leader selection.



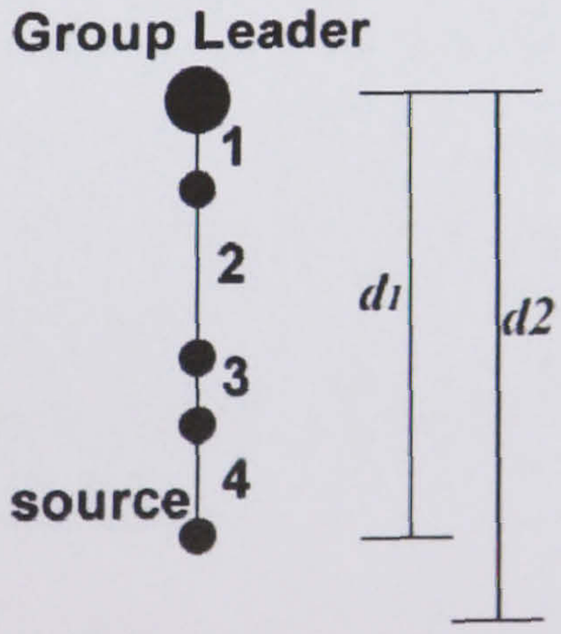
Case 1. Fixed Radio Range



Case 2. Adjusted diffrent radio range for diffrent nodes



Case 1



Case 2

Figure 4-4: Distance or number of hops from a source to its group leader.

4.5.4 Layer 4: Key Management for MSNs

There are many complicated research issues relating to MSNs, including localizations, routing, network management, topology maintenance, security and many more. As we have seen, WSNs have many applications all with different requirements and different levels of available resources. There can be many different scenarios in which MSNs are used. In some cases all sensor nodes may be mobile, whereas in other cases there may be some static sensors and some mobile. Furthermore there can be different types of roaming for mobile sensor nodes, including free roaming (e.g. applications like MSNs in water) and guided roaming (e.g. applications in which sensor nodes attached to reboots, human body and vehicle). These properties of MSNs increase the challenges compared to static WSNs. Furthermore if we consider scalability issues with MSNs, things become more complicated again. Secure communication is also an essential requirement of MSNs, in similarity with other wireless networks. We intend to propose a secure communication solution for MSNs, but to achieve this we must also consider non-security related issues (such as scalability) which have a significant influence on MSN security.

In this section we describe our proposed key management solution for MSNs, according to network available resources and security requirements. This key management solution is based on our proposed protocol SADI-GKM (Structure And Density Independent Group-Based Key Management) Layer 1, Layer 2 and Layer 3. In this proposed key management solution the MSN uses SADI-GKM Layer 1 for basic key establishment, Layer 2 for secure data aggregation, Layer 3 for secure group leader selection and a key pool based key management process for mobile sensor node authentication within the host group. Furthermore we use the micro-mobility concept for our MSN key management scheme. We assume that every sensor node has stored its position and the boundary coordinates of the group. The boundary coordinates of the group can help the mobile sensor to establish when they enter a neighbouring or other group territory. We will describe this scheme in detail in chapter five and six.

4.6. Summary

In this chapter we have presented design overview of our novel protocol SADI-GKM. This protocol combines four different layers with different schemes. All these layers are integrated with each other to provide better secure solutions for multiple proactive security issues. The first layer of this protocol provides basic key management to all sensor nodes. The second layer's responsibility is to provide secure data aggregation according to the need of target applications. The third layer provides facilities for the selection of a new aggregator or group leader sensor node. This layer deals with the tasks of how to initiate the selection process, how to gather information from member sensor node, and how to find the next suitable group leader sensor node. The fourth layer's main task is to provide key management services for mobile sensor nodes in a WSN. The reason for the inclusion of this layer is due to its inter-relationships with the issues addressed by the other layers. Ignoring any such issue can weaken the network security. We have also identified requirements, issues and challenges that are important when designing an effective key management protocol for large scale WSNs.

Chapter Five: SADI-GKM Protocol Components

In the previous chapters we have described the basic functionalities of our protocol. In this chapter we explain the operation of all the protocol layers in detail. Before going into the protocol design details, we describe some interesting parts of our project development, including some key research investigations that have helped us to design an efficient protocol. These key analyses are described in Section 5.1. For example we have established three important parameters during our analysis of node capture attacks as to be explained in Section 5.1.1. Unless balanced carefully, these three parameters can help an adversary to compromise the communications of an entire WSN. Similarly we have also investigated how a group leader's position in a group can affect the performance of the entire group of sensor nodes, as will be described in section 5.1.2.

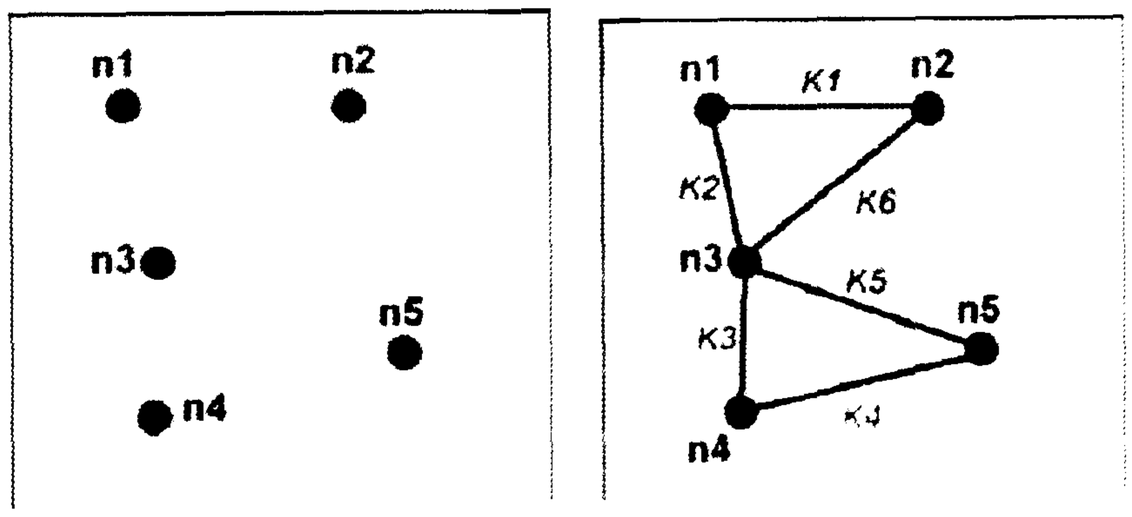
We then continue our discussion with the key management and secure data aggregation layers, where we explain about different algorithms, and how we can perform the secure data aggregation and key management tasks together. This constitutes part of our novel SADI-GKM protocol. Next we explain our novel secure group leader selection scheme which comes under the third layer of our protocol. Finally, we discuss key management for MSNs which come under the fourth layer of our protocol.

5.1 SADI-GKM Pre-design Investigations

5.1.1 Node Capture Attacks

Node capture attacks are one of the most important and distinct issues that catch the attention of many researchers. Therefore a number of different key management schemes have been proposed as described in Chapter three. As a part of our investigation on node capture attacks we proposed a simple protocol called Dynamic Group-Based Key Establishment (DGKE) [118, 33, 163, 71].

In DGKE we assign a unique master key to each group of sensor nodes. These keys will be set up in each sensor node before deployment. Once deployed, sensor nodes in a group will generate new keys using the master key and a random number. The sender sensor node will encrypt a *hello* message using this newly generated key and send the resulting ciphertext to its neighbours along with the random number used to generate the key. The neighbouring sensor nodes will use the master key and the received random number to decrypt the ciphertext message. Once decryption is successfully completed that key will be used for further communication. The sender and receiver sensor nodes will perform these steps once after deployment, after which the non-group leader nodes will no longer retain the master key. The group leader retains the master key for use when new nodes join or leave the network [33]. We can explain this process with an example. We assume Figure 5-1 (a) as a group of five sensor nodes: n1, n2, n3, n4 and n5. All these sensor nodes have a shared key K (the master key).



(a) (b)

Figure 5-1: (a) Nodes deployed. (b) Key establishment.

After successful sensor nodes deployment, each pair of nodes generate a key using the master key $\underline{K}$ (shared between all the group member sensor nodes) and a random number r (generated by the initiator node and sent to the other node), i.e. K_n = hash (K, r) as shown in Figure 5-1 (b). Specifically, the generated keys are defined below:

K_1 = hash (K, 5) → shared between n1 and n2

K_2 = hash (K, 65) → shared between n1 and n3

K_3 = hash (K, 31) → shared between n2 and n4

K_4 = hash (K, 57) → shared between n4 and n5

K_5 = hash (K, 52) → shared between n3 and n5

K_6 = hash (K, 101) → shared between n2 and n3

These keys will be generated once and only for the nodes within the group.

(a) Analysis

We have simulated DGKE against node capture attacks using various different topologies to find the core properties which can help an adversary to compromise the communication of an entire sensor network. Our main goal in discovering these characteristics is that they can be avoided during the design of further WSN protocols. In this section we describe the detailed process of how we design our simulation framework and how we conduct the simulations.

The simulation process involves two parts: building an appropriate test topology, and measuring a sequence of sensor node compromises to establish the effects that they have on the security of the WSN. We describe each of these parts separately. In the first part we describe how the topology creation process is performed. In the second part we describe the process of node capture attacks on the WSN. Using these simulations we intend to find the probability of total communication compromise after a certain number of node capture attacks have taken place. This is achieved by randomly selecting a sensor node during the simulation and assuming that this node has been physically compromised, after which the affects on the network of this compromise are measured.

We performed a number of tests for comparison, pseudo randomly compromising 100, 200 and 300 sensor nodes of the entire WSN. Even though a large scale WSN might be expected to have many more sensor node performing the test in this way allows us to provide a suitable comparison to analyse the effect of node capture attack on different topology. In every simulation for tree, grid and random mesh topologies we count the total number of communication links in the entire WSN and count the compromised links after having launched the stated number of node capture attacks. These two values give

us the proportion of compromised communication for the entire WSN. We performed 5000 runs of the simulation for every set of node capture attacks to determine the average proportion of compromised nodes for a particular value *n* of node compromises. This provides us with a probability for the likelihood of a communication link being compromised after a certain number of sensor nodes within the WSN have been compromised. For all of these simulations, we assume that if the group leader sensor node is compromised the communication for the entire group becomes compromised because the node holds the master key. In a Heterogonous WSN group leader might have different characteristics from member sensor nodes. However we assume the likelihood of compromising the group leader node will be the same as other member sensor nodes.

(I) Node capture attacks on tree topology

As discussed earlier we organise the entire WSN into groups. Each group has 100 sensor nodes and is structured as a tree topology. Each sensor node is connected to its parent and between 1 and 5 child sensor nodes. The number of child sensor nodes for each parent sensor node is selected randomly. Figure 5-2 (a) shows 100 sensor nodes organised in such a tree topology.

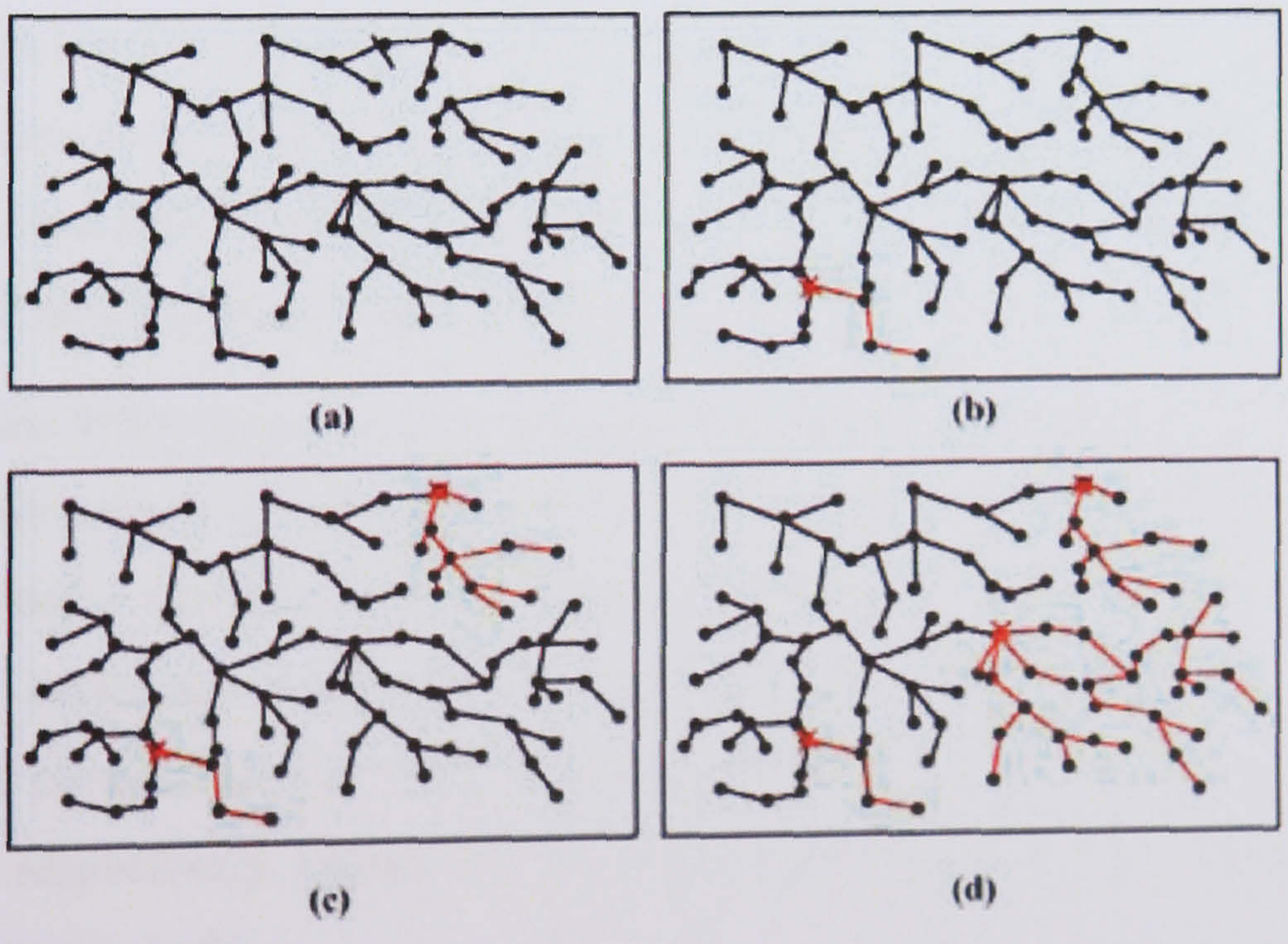


Figure 5-2: (a): 100 sensor nodes in random tree topology. (b): Compromised links after the first random node capture attack. (c): Compromised links after the second node capture attack. (d): Compromised links after the third node capture attack.

In the simulation process we assume every child sensor node shares a key with its parent sensor node. We assume that all keys will be successfully established using the process described in Section 5.1.1. After successful establishment of the WSN in a tree topology formation, we launch several node capture attacks. For each attack we randomly select a node and assume that it has been physically compromised, namely, all its stored data are known to the attacker. Since the data include encryption keys, all the communication links of the compromised sensor node will also be compromised. Moreover if the compromised sensor node is a parent node, all data sent by its child sensor nodes to the sink must pass through it, and consequently the communications links of all of these child sensor nodes should also be considered compromised. Figures 5-2 (b), (c) and (d) show the effects of one, two and three node capture attacks at different positions within the group respectively. The crosses indicate the compromised sensor nodes and highlighted lines between sensor nodes indicate compromised communication links resulting from the compromise of these sensor nodes.

(II) Node capture attacks on grid topology

This process is used to organise the entire WSN in a grid topology formation. We follow a similar process to that of the tree topology, managing the entire WSN in groups of 100 sensor nodes. In a grid of 100 sensor nodes every sensor node has a maximum of 4 and a minimum of 2 neighbouring nodes, as shown in Figure 5-3 (a).

Once the entire WSN has been established we start the testing process. This is achieved similarly to the process for the tree topology where we randomly capture nodes. Once a node is physically compromised, we assume that all the communication links with its neighbouring nodes are compromised as well. Figures 5-3 (b) and (c) visualize the compromised communication links after the physical compromises of two and four sensor nodes respectively. Figure 5-3 (d) shows this effect after the compromise of the group leader sensor node.

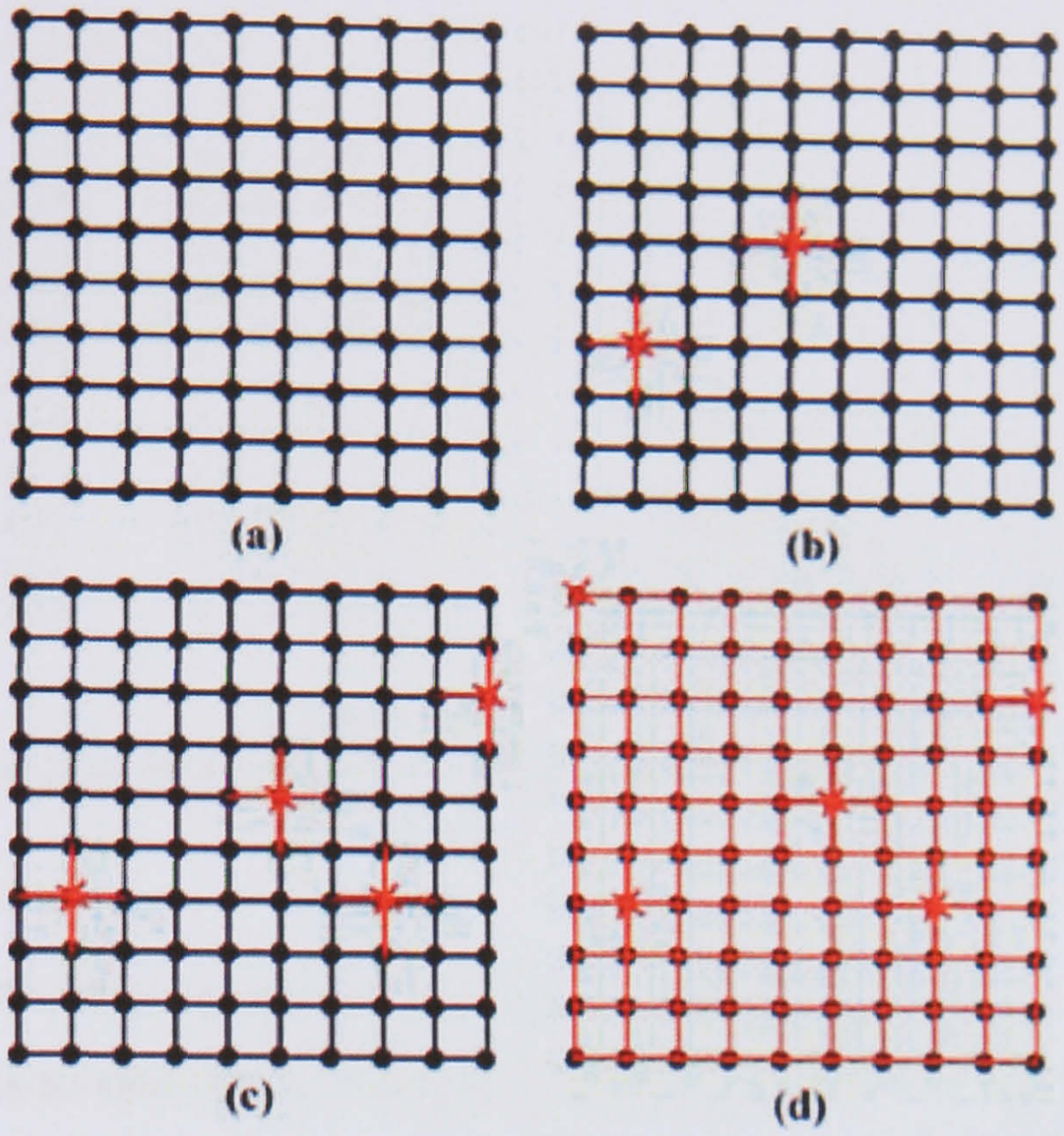


Figure 5-3: (a): 100 sensor nodes in a grid topology. (b): Compromised links after two random nodes capture attacks. (c): Compromised links after four node capture attacks. (d): Compromised links after the compromise of the group leader node (top left sensor node).

For a better understanding of the process Figure 5-4 presents a further depiction of the compromised and uncompromised keys after *n* node capture attacks. The numbers indicate a sensor node's complement of *uncompromised* keys and the highlighted zeros therefore indicate the location of the compromised sensor nodes. For example, in Figure 5-4, after the first attack one sensor node at location (7, 4) is compromised, resulting in the compromise of the four shared keys with the four neighbouring sensor nodes. The compromised node therefore holds no uncompromised keys, while the four neighbouring sensor nodes each hold three uncompromised keys. Note that all communication links of a compromised sensor node with its neighbouring sensor nodes are compromised. Therefore for each compromised neighbour we subtract by one the number of uncompromised links from neighbouring sensor nodes. For each simulation we count the total communication links compared against the compromised communication links to find the probability for certain number of attacks.

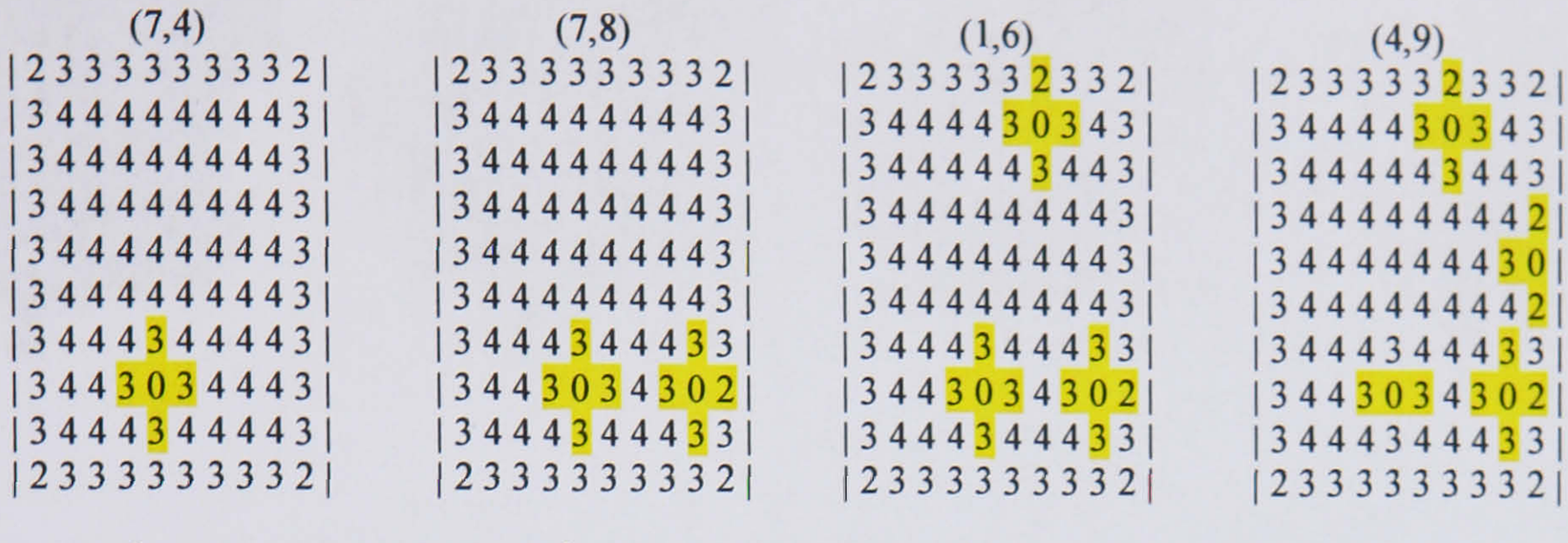


Figure 5-4: Random four node capture attacks in a group.

At every simulation step we count the total communication links and compromised communication links to find the probability of communication compromises given certain number of attacks. We ran this process 5000 times to get an averaged probability figure.

(III) Node capture attacks on random mesh topology

The setup of the random mesh topology is different from those of the tree and grid topologies. In a random mesh topology we select neighbouring sensor nodes on the basis of radio ranges between sensor nodes. Effectively, all sensor nodes coming within a sensor node's radio range will form its neighbours. Nodes in every group are randomly deployed and assigned geographical coordinates (x, y). We assume that the radio range of every sensor node is 40 meters. The geographical area of each group is 150×150 meters, containing 100 sensor nodes. Every sensor node in a group will establish communication links with its neighbouring sensor nodes falling in the range of 40 meters. Figure 5-5 (a) shows the random mesh topology in a group.

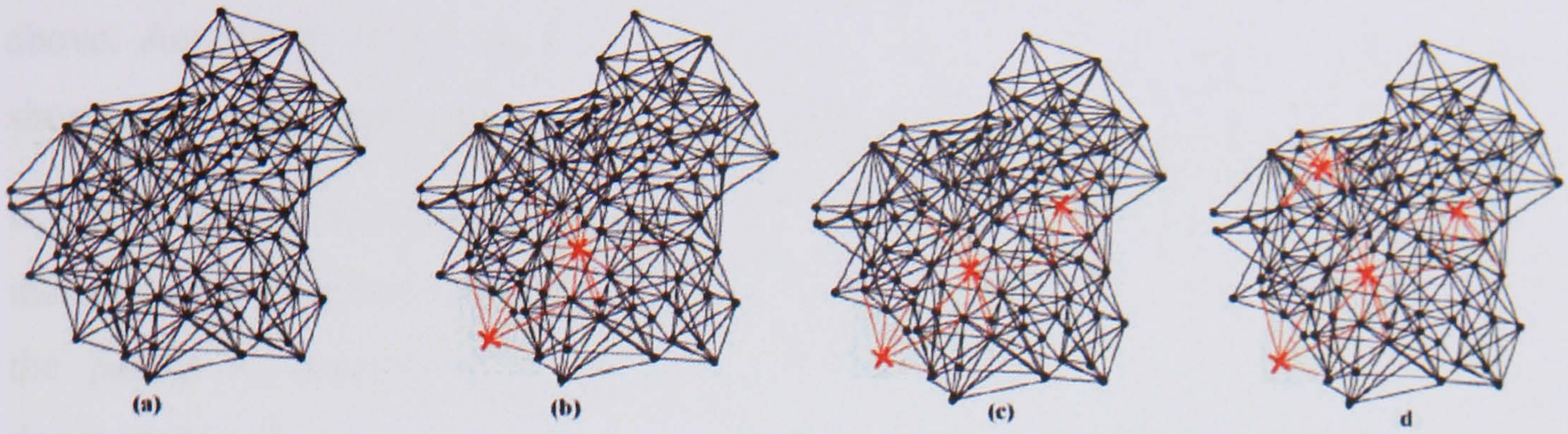


Figure 5-5: (a): 100 sensor nodes in a random mesh topology. (b): Compromised links after two random node capture attacks. (c): Compromised links after three node capture attacks. (d): Compromised links after four node capture attacks.

As explained in Section 5.1.1, using DGKE every sensor node will store the number of keys according to its local neighbourhood density. Consequently if a sensor node has 10 neighbouring sensor nodes, it will store 10 different keys. In a random mesh topology we select the number of neighbouring sensor nodes on the basis of radio ranges and therefore the number of keys stored in each sensor node will vary, in contrast to the situation with the grid topology.

The simulation process for the random mesh topology is similar to those of the tree and grid topologies. Figures 5-5 (b), (c) and (d) show two, three and four node capture attacks at different positions within the group respectively. As described earlier, the crosses show the positions of compromised nodes and the highlighted lines between sensor nodes show compromised links. To find the probability of the total communication compromised we count the total communication links and compromised ones, thereby establishing the effect of node capture attacks.

Conclusions

Figure 5-6 shows the effect of node capture attacks on the tree, grid and random mesh topologies using groups, based on the results obtained from the simulations described

above. According to our simulation results, a WSN using the random mesh topology shows better resilience against node capture attacks than the grid and tree topologies.

It is clear that the node capture attacks are more dangerous for the tree topology than the mesh topology because the tree topology has only one route from a child to its parent. If the parent is compromised, all data passing through the node from its children downwards will also be compromised. The results show a high risk associated with the use of tree topologies as compared to other topologies in terms of node capture attacks. New and improve security protocols for tree topology might improve security but with high cost which can be easily achieve using other topologies like grid and random mesh.

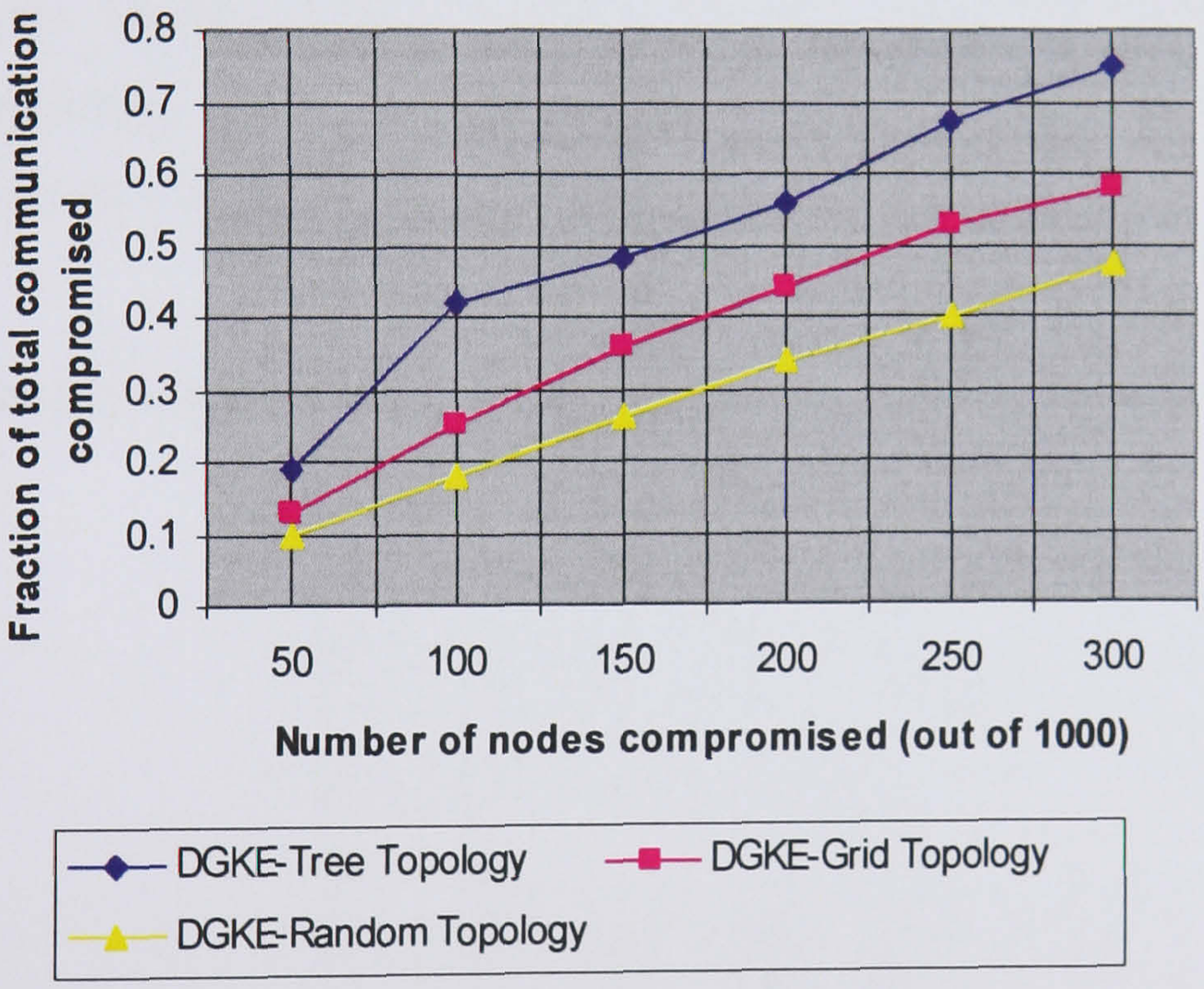


Figure 5-6: Node capture attack on DGKE using different topologies.

On the basis of these simulation results we have established useful information concerning the nature of node capture attacks. Therefore during the design of our SADI-GKM protocol we have attempted to avoid sharing keys which are used for encryption and decryption, chosen appropriate neighbouring node densities, and been careful to allow the selection of an appropriate topology.

5.1.2 Importance of Group Leader Positions

In this section we propose a formula for new group leader selection. This formula considers four different selection parameters: the energy of a candidate sensor node, the number of its neighbouring sensor nodes, the trust level of the sensor node and the position of the sensor node. As sensor nodes' energy is limited and they will stop working after finishing their available energy, it is understandable why energy is an important parameter in the group leader selection. Similarly the fewer the neighbours a sensor node has, the less suitable the sensor node is to be selected as a new group leader Furthermore a sensor node with low trust should also be avoided. All these parameters are very important in the selection process of a new group leader. However to verify the importance of the fourth parameter "position of a new group leader", we found it necessary to undertake some investigative analysis. This analysis, which we outline below, has shown us the importance of sensor node positions in the selection of a new group leader.

In this evaluation process we have assumed a simple scenario as shown in Figure 5-7.

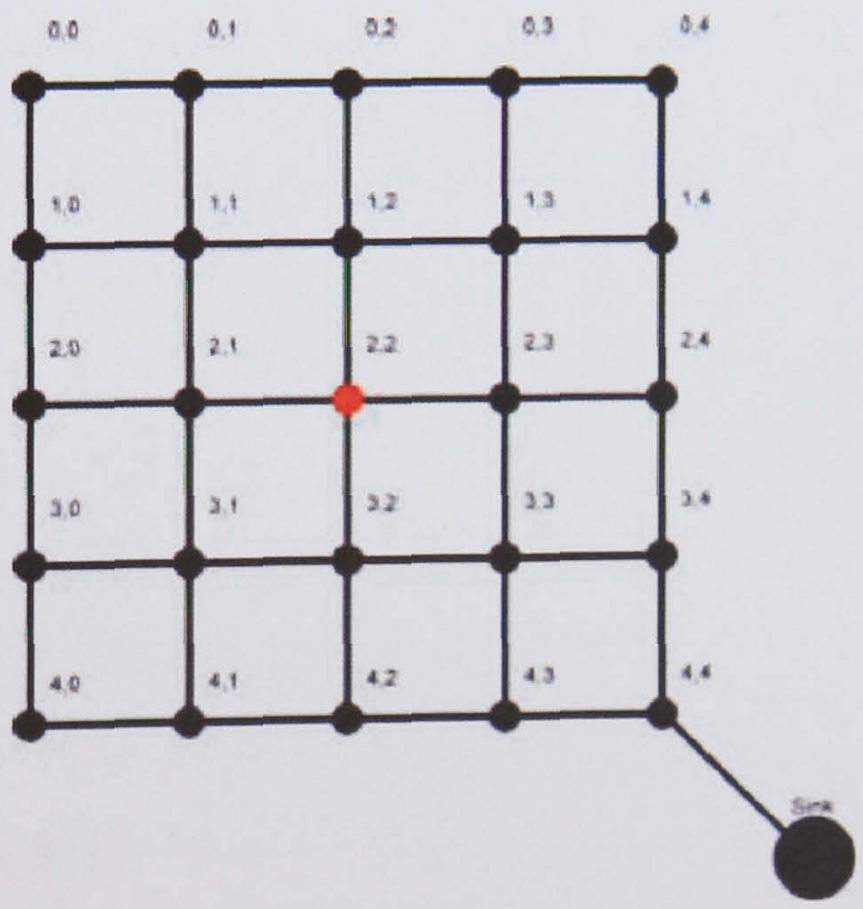


Figure 5-7: Group leader positions and communications towards the sink.

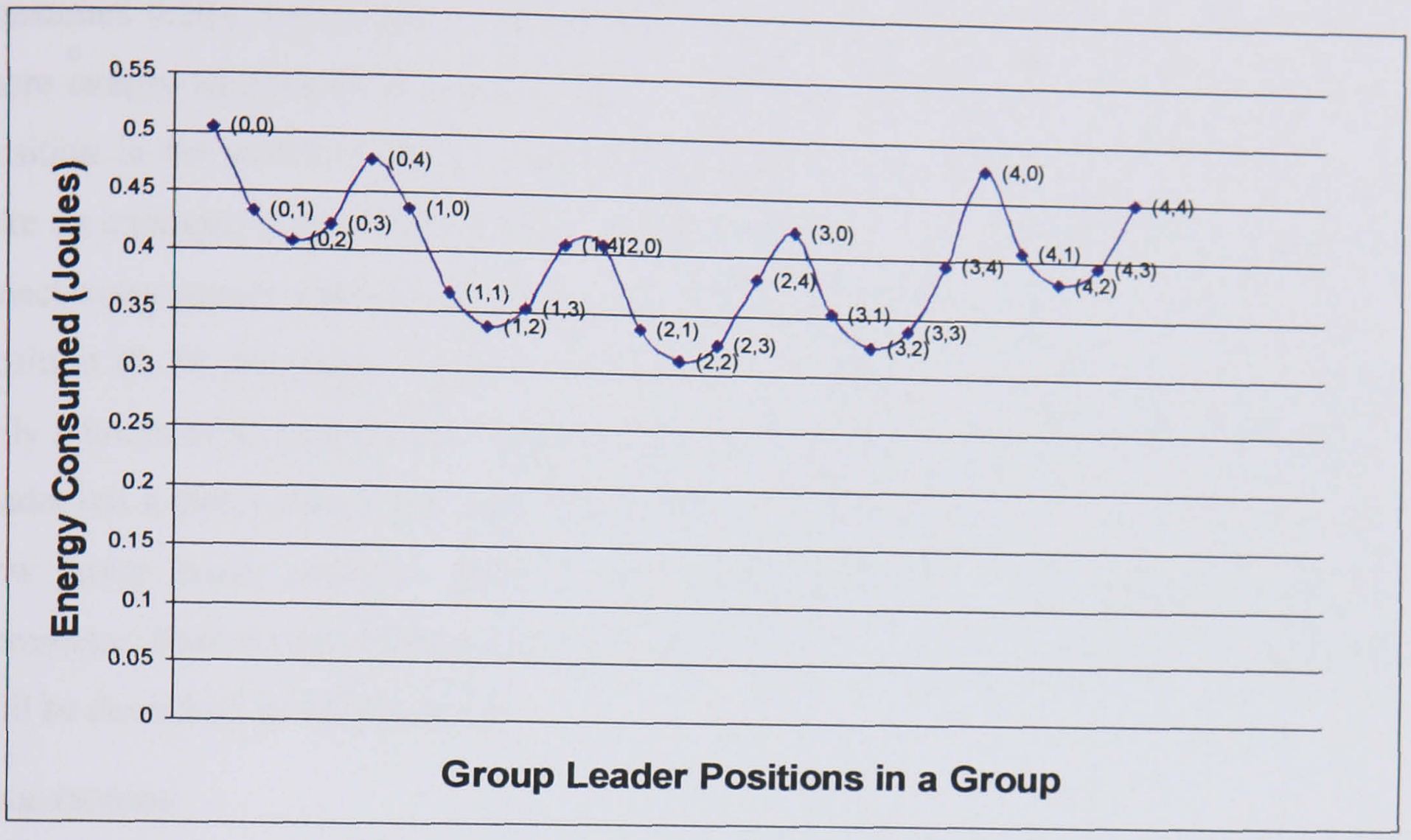


Figure 5-8: Energy consumptions of a group with different positions for the group leader.

During this simulation we have selected a group of 25 sensor nodes in a grid topology as shown in Figure 5-7. Each sensor node has one joule of energy. The packet size is 48 bits. We have placed the group leader sensor node at all possible positions in the group. In this evaluation process all sensor nodes sense and route data towards the group leader using an adaptive routing algorithm [162]. The group leader receives data from 24 nodes and aggregates all the data before finally routing it towards the sink. This entire process is considered to be one cycle. We measure the energy consumption of every single node in Joules using *first radio model* described in Section 6.2, phase 3, which calculates the cost of transmitting and receiving packet of size *k* bits. These energy outcome values might be different in real applications where different sensor nodes are used with different radio models. We have run 100 cycles for each group leader and then calculated the energy consumption of the entire group as shown in Figure 5-8. We have also run the same simulation for a group of 100 sensor nodes, providing us with a large amount of useful information. In particular the sensor at position (2, 2) is the most efficient which consume

0.310 Joules in comparison to the least efficient sensor node at position (0, 0) which consumes 0.504 Joules which means the sensor node at position (0,0) consumes 20% more energy as compared to position (2, 2). As shown in Figure 5-7, the group leader position in the middle helps to save energy consumptions. For better understanding we take an example if the battery life of every sensor node in a group is 10 hours and we select every sensor node in the group as group leader for 100 cycles. The sensor node at position (0, 0) consumes 7 hours of its battery life where as node at position consumes only 5 hours of its battery life. These results clearly indicate that the position of the group leader has a direct affect on the performance of the entire group. Based on this, during the new group leader selection process we should use the node positions as a selection parameter. Further details about our proposed solution for the new group leader selection will be described in Section 5.2.2.

Conclusions

The outcome and results from our experiments show the importance of the group leader`s position and its effect on the WSN performance. This importance will increase further as the size of the group increases, or as the packet size increases. With large group or packet sizes, the inappropriate selection of a new group leader can result in a large energy overhead.

5.2 SADI-GKM Protocol Components

This section presents our main protocol (SADI-GKM). To apply the SADI-GKM technique to a WSN, a number of assumptions are necessary in order to provide a consistent framework within which to work. It will also be useful for us to consider the requirements of our design. These are outlined in the following subsections.

5.2.1 Key Management and Secure Data Aggregation (First and Second Layer of SADI-GKM)

Generally WSNs are scalable networks, and it is not uncommon for them to incorporate thousands or even millions of sensor nodes. Researchers have proposed a number of different network management protocols and schemes for large scale WSNs. A common idea proposed for the management of large-scale networks is that of splitting them into regions or small groups of nodes (logically or physically) using clustering, geographical division, topology *etc.* To deal with the scalability issue we therefore organise large-scale WSNs into small groups of sensor nodes with unique IDs.

As for deployment, a Gaussian distribution can be used to establish a random group deployment of sensor nodes in some outdoor applications [78]. For indoor applications sensor nodes can easily be deployed in groups. However, we do not intend to consider the issue of node deployment in detail at this stage. The notations to be used for the protocol presentation are summarised as follows:

GI	Group ID
NI	Node ID in group
ID	Node ID in network (concatenation of GI and NI)
M_{kGI}	Symmetric master key shared between sink SI and group leader GI
M_i	Encrypted data of node ID_i
M_{GI}	Encrypted data of group leader node GI
$K_{SI,IDi}$	Symmetric key shared between node ID_i and the sink SI. This key is used for the encryption and decryption of data collected by ID_i
$K_{GI,IDi}$	Symmetric key shared between node ID_i and its group leader node GI
V_{GI}	Key/secret shared among all the sensor nodes in group GI
$U_{a,b}$	Key/secret shared only between neighbouring group leaders a and b
TS_{IDi}	Time stamp for node ID_i
TS_{GI}	Time stamp for group leader node GI
d_i	Sensed data by node ID_i
X_i/Y_i	Hash values calculated by sensor node/group leader node

(a) Key Pre-distribution Phase

In the first phase of key pre-distribution the following steps are performed. This occurs before the deployment of the sensor nodes in groups:

- Assign a global unique ID to each sensor node. This global ID is comprised of a group ID, *GI* and an ID within the group, *NI* (see Figure 5-9).
- Assign a unique group master key M_{kGI} to every group leader in the WSN.
- Assign a unique value V_{GI} to each group of sensor nodes, used for authentication inside the group.
- Assign a unique value $U_{a,b}$ to every pair of neighbouring group leader sensor nodes.
- Generate and store a unique key $K_{GI,\ IDi}$ using the corresponding group master key M_{kGI} and a sensor node ID for every group member sensor node.
- Assign a unique key $K_{SI,IDi}$ to every sensor node ID_i (note: this key need only be assigned if we require the aggregation of data in an encrypted form at group leader sensor nodes).

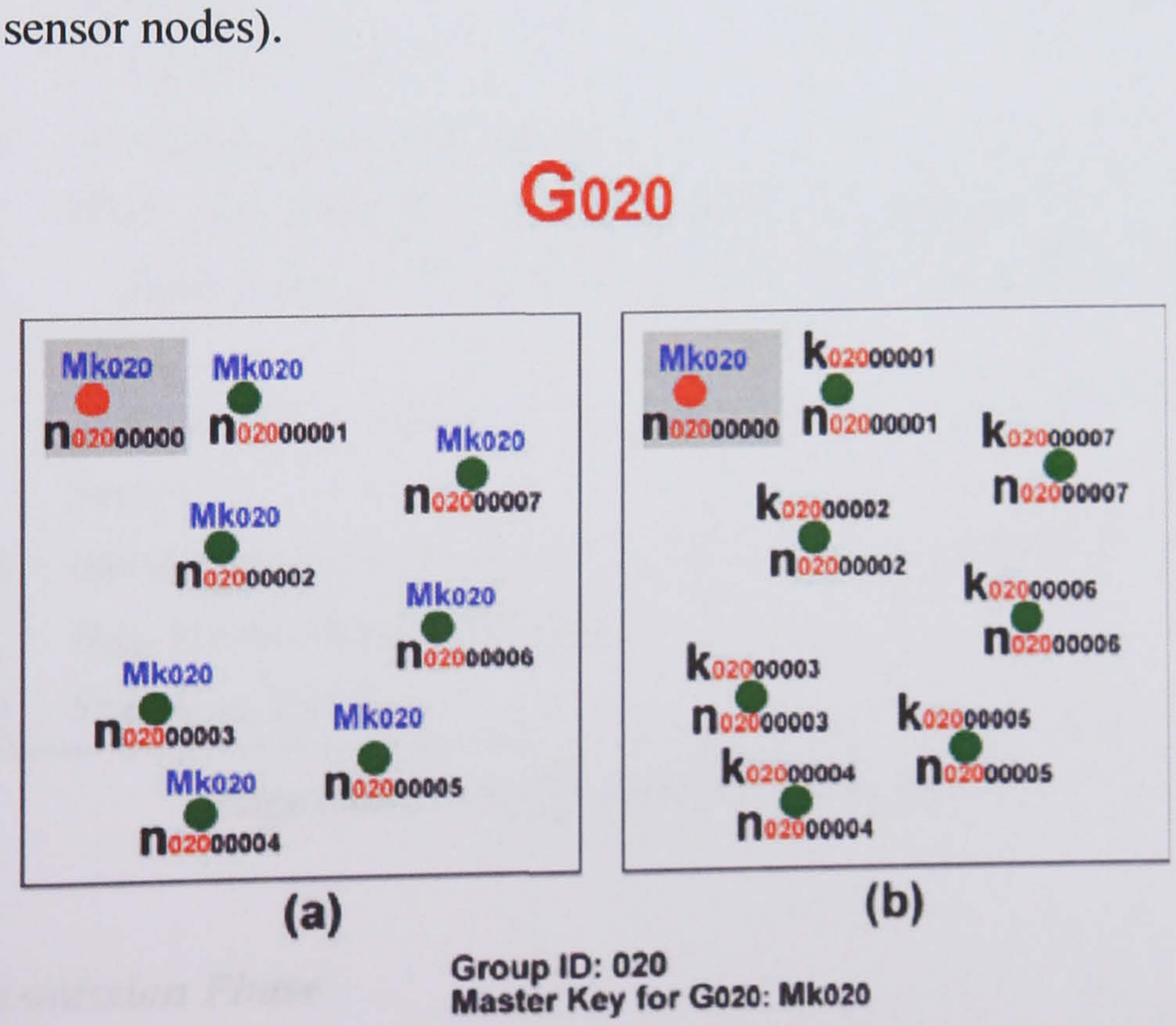


Figure 5-9: Organization of sensor groups.

In the example illustrated in Figure 5-9, a sensor node has a global ID of the form **02000002**, with the initial three digits representing *GI* (group ID) and the last five digits representing *NI* (local node ID in a group *GI*). Algorithm 5-1 shows the complete pre-distribution process. According to step 2 there will be no master key stored in any member sensor nodes. These steps should be performed by an administrator prior to deployment. It is important to emphasise that sensor nodes in all groups will use different keys $K_{GI,ID}$ for encryption purposes in order to provide better confidentiality (as discussed in section 5.1.1). The reason for keeping the master key M_{kGI} at the leader of each group *GI* is to allow it to decrypt the encrypted data received from the member sensor nodes in its group. In every single group data will be encrypted at source sensor nodes and is only decrypted at the group leader node for data aggregation purposes, as will be explained in later sections.

– **Pre deployment** –

Step 1: *Node ID formation*

Concatenation of Group ID and Node ID forms a unique ID for each node in the Network

ID ← join (*GI* , *NI*)

Step 2: *Group key generation for node ID*

If *ID* = join (GI, 00) then //Check if the node is a group leader

$K_{GI,ID} \leftarrow M_{kGI}$ //Assign a group leader master key M_{kGI} to the node

Else

$K_{GI,ID}$ ← hash (M_{kGI}, *ID*) //Assign a group member key to the node

End if

Step 3: *Sink key generation for node ID only if secure data aggregation is required*

$K_{SI,ID}$ ← *a key shared with the sink*

Step 4: *Store* $K_{GI,ID}$, V_{GI}, $K_{SI,ID}$

Algorithm 5-1: Pre-distribution phase.

(b) Data Transmission Phase

We first discuss the functions of each non-group-leader sensor node, which are defined in Algorithm 5-2 as part of the proposed protocol. Suppose that after successful deployment of the sensor nodes, an event occurs. In the step 1 of the *Send* function in Algorithm 5-2.

an event source node with its identity ID_i will collect and encrypt the event data d_i with its key $K_{SI,IDi}$ shared with the sink to produce C_i. To enable data freshness checking by the group leader of node ID_i for the detection of a replication attack, we produce a value M_i by re-encrypting C_i with a time stamp TS_{IDi} using the key $K_{GI,IDi}$ shared between node ID_i and its group leader GI, where "$C_i \mid TS_{IDi}$" in $E_{KGI,IDi}(C_i \mid TS_{IDi})$ signifies the concatenation of C_i and TS_{IDi}.

In step 2, node ID_i produces an authentication code X_i by hashing M_i, the key/secret V_{GI} shared among all the group members, ID_i and TS_{IDi}, for communication with neighbouring sensor nodes. After finding a correct neighbouring sensor node using its energy level and direction value, sensor node ID_i sends out (X_i, M_i, ID_i, TS_i). The value X_i will be used by the receiver sensor node to authenticate the received information and TS_{IDi} is used to avoid replay attacks.

When the neighbouring sensor node receives the information, it produces an authentication code X_j by hashing the received items (M_i, ID_i, TS_{IDi}) with shared key V_{GI}, and compares it against X_i for the purposes of authentication, as illustrated in the *Receive* function of Algorithm 5-2. If successful, the received data are forwarded to the next neighbour. In this way, we avoid encryption and decryption at every single hop to reduce the usage of limited recourses available to the nodes.

– Group member nodes –

Send () { // Works inside a group

Step 1: *Collect, encrypt and assign sensed data to variable* M_i

$d_i \leftarrow$ Sensed_data ()

$C_i \leftarrow E_{K_{SI,IDi}}(d_i)$

$M_i \leftarrow E_{K_{GI,IDi}}(C_i \mid TS_{IDi})$

Step 2: *Send out encrypted data* M_i, *hash value* X_i, *source node* ID_i, *and time stamp* TS_{IDi}

$X_i \leftarrow hash\ (M_i, V_{GI}, ID_i, TS_{IDi})$

Return $(X_i, M_i, ID_i, TS_{IDi})$

}

Receive (X_i, M_i, ID_i, TS_i) {

Step 1: *Authenticate the sender node*

$X'_i \leftarrow hash\ (M_i, V_{GI}, ID_i, TS_i)$

If $X'_i = X_i$ and TS_i is fresh then

Step 2: *Forward received data to next neighbour towards group leader*

Return (X_i, M_i, ID_i, TS_i)

Else

Abort and report to group leader GI

}

Algorithm 5-2: Group member nodes.

Second, each group leader sensor node has additional responsibilities, including secure data aggregation, integrity checking and authenticate communication with other group leaders. Algorithm 5-3 illustrates the functions performed by each group leader sensor node.

As shown in the *Receive* function of Algorithm 5-3, when a group leader sensor node receives data, it will first check whether the data are from another group leader node or a member node, based on two calculated hashes X_{GI} and X_i. Here, secrets V_{GI} and $U_{a,b}$ used for the calculation of the hashes are stored in the sensor node prior to its deployment. In the case where the data are from a member node (i.e. $X_i = Y_i$), the leader sensor node applies its master key M_{kGI} and received identity ID_i to compute shared key $K_{GI,IDi}$ for the

decryption of M_i to recover C_i and TS_{IDi}. It saves C_i if TS_{IDi} is fresh, and discards C_i otherwise. The successful decryption of M_i can also be used to authenticate source sensor node ID_i and check data integrity. After the data from all expected group member sensor nodes have been received, the leader sensor node invokes the *Send* function of Algorithm 5-3.

If the data is from another group leader (i.e. $X_{GI} = Y_i$) and TS_i is fresh, then the received data are forwarded to the next appropriate group leader on the route to the sink.

We intend to apply SADI-GKM to different applications. Therefore for the secure communication between groups, we can establish key management according to the structure of applications. For example, we adopt the assumption used in LEACH [139], namely, each group leader node has a larger communication range than an ordinary one so that neighbouring group leader nodes can communicate with one another directly. This assumption can also be considered in order to allow non-group-leader sensor nodes to mediate communications between group leaders.

– Group Leader Nodes –

Send () {

Step 1: *Compute aggregated data D_{GI} from encrypted data, C_1, ..., C_n, received from group member nodes*

$$D_{GI} \leftarrow (\sum_{i=1}^{n} C_i) \bmod Z \qquad 0 < D < Z$$

Step 2: *Encrypt D_{GI} and TS_{GI} to produce M_{GI}*

$M_{GI} \leftarrow E_{M_{kGI}}(D_{GI}, TS_{GI})$

Step 3: *Send out encrypted data M_{GI}, hash value Y_{GI}, group identity GI, and time stamp TS_{GI} from group leader sensor node a in group GI to group leader b*

$Y_{GI} \leftarrow hash\ (M_{GI}, U_{a,b}, GI, TS_{GI})$

Return (Y_{GI}, M_{GI}, GI, TS_{GI})

}

Receive (Y_i, M_i, ID_i, TS_i) {

Step 1: *Check if the data are coming from a member node or another group leader*

$X_{GI} \leftarrow hash\ (M_i, U_{a,b}, ID_i, TS_i)$

$X_i \leftarrow hash\ (M_i, V_{GI}, ID_i, TS_i)$

If $X_i = Y_i$ then // Data from a member node

Step 2: *Decrypt M_i using key $K_{GI,IDi}$ where $K_{GI,IDi} = hash(M_{kGI}, ID_i)$, to recover C_i and TS_{IDi}*

$(C_i, TS_{IDi}) \leftarrow E^{-1}{}_{K_{GI,IDi}}(M_i)$

If TS_{IDi} is fresh then

Save C_i

If C_i is the last value received then

Step 3: *Send out aggregated data*

Send ()

Else

Discard C_i and report to sink

Else if $X_{GI} = Y_i$ and TS_i is fresh then

Step 4: *Forward the data received from a group leader to another group leader towards the sink*

Return (Y_i, M_i, ID_i, TS_i)

Else

Abort and report to sink

}

Algorithm 5-3: Group leader nodes.

For the *Send* function of Algorithm 5-3, the encrypted data items $\{C_1,\ldots\ldots,C_n\}$ are first aggregated to produce D_{GI} using homomorphic encryption as described in section 3.2. We note that the value of Z must be chosen large enough to prevent overflow. As with Algorithm 5-2, it is necessary to encrypt D_{GI} and a time stamp TS_{GI} with the key M_{kGI}, shared with the sink, for the purposes of data authentication and freshness checking. The *Send* function also uses the secret $U_{a,b}$, shared among a pair of neighbouring group leader sensor nodes a and b, and a time stamp TS_{GI} to produce an authentication code Y_{GI} to deter replay attacks. Finally, the data items (Y_{GI}, M_{GI}, GI, TS_{GI}) are sent to the first neighbouring group leader sensor node on the route to the sink.

Third, Algorithm 5-4 explains the decryption process performed by the sink. We assume the sink has all the keys $U_{a,b}$, M_{kGI} and $K_{SI,IDi}$ shared with the group leaders and sensor nodes. Once the data have been received at the sink it will go through a process to verify the source group leader (i.e. by checking that $Y_{sink} = Y_{GI}$ and TS_{GI} is fresh). However the Y_{GI} can be forged if a group leader node is compromised. Step 2 ensures that the time stamp and data have not been altered by a compromised group leader (i.e. $E^{-1}{}_{MkGI}(M_i)$ and $TS'_{GI} = TS_{GI}$) otherwise the sink should discard the received information and take appropriate actions. If step 1 and 2 are successful then step 3 will be performed to retrieve the aggregated data as shown in Algorithm 5-4.

– Sink Node –

Receive (Y_{GI}, M_{GI}, GI, TS_{GI}) {

Step1: *Authenticate the sender group leader*

$Y_{sink} \leftarrow$ hash (M_{GI}, $U_{a,b}$, GI, TS_{GI})

If $Y_{sink} = Y_{GI}$ and TS_{GI} is fresh then

Step2: *Decrypt M_{GI} using master key M_{kGI} to recover D_{GI} and TS_{GI}.*

$(D_{GI}, TS'_{GI}) \leftarrow E^{-1}_{MkGI}(M_{GI})$

If $TS'_{GI} = TS_{GI}$ then

Step3: Get the aggregated data

$$S_x \leftarrow (D_{GI} - (\sum_{i=1}^{n} K_{SI,IDi})) \bmod Z$$

Else

Abort and take actions

}

Algorithm 5-4: Sink node.

The sink can calculate the average μ as $\mu = S_x/n$. To calculate the variance σ^2 every sensor node needs to send the square of each sensed value to its group leader so the variance can be calculated as $\sigma^2 = S_x/n - \mu^2$ where S_x is the sum of the squared value of each sensor node. If the group leader node is interested to find the minimum or maximum values from received encrypted values, we can use the Order Preserving Encryption Scheme for Numeric Data (OPES) [164]. This scheme allows comparisons to be directly applied to encrypted data. Existing results have shown that homomorphic encryption is a cost effective solution for secure data aggregation in sensor networks [165].

5.2.2 Secure Group Leader Selection (Layer three)

In this section we propose a formula for new group leader selection. In this formula we consider four important factors in the selection process. These factors include available energy, trust level, number of neighbouring nodes and position of a new group leader node. During the selection process the available energy E_l and trust level T_l need to be

greater than the thresholds δ and α respectively, to become eligible for new group leader selection.

This formula can be used for different applications with respect to different communication behaviours of sensor nodes. For example in some applications sensors might use different radio ranges with respect to distances between different sensor nodes as described in the previous chapter. In such applications we measure the space S_l between source sensor nodes and the current group leader using distance d_l. However in applications where a fixed radio range is used for communication between sensor nodes, we measure the space S_l between source nodes and the current group leader using hops h_l. Furthermore our proposed formula can also be used in heterogeneous WSNs where different sensor nodes have different energy requirements. However, existing solutions for group leader selection/election have only used energy or the number of neighbouring sensor nodes as their main election criteria [140, 160, 139, 161].

We have used the following notation in our proposed formula:

S_l	Space between source sensor node l and the group leader.
d_l	Distance between sensor node l and the group leader.
h_l	Number of hops between sensor node l and the group leader.
δ	Energy threshold level.
α	Trust threshold level.
w_e	Weighting factor for energy.
w_{sl}	Weighting factor for the space between sensor node l and the group leader node, where w_{sl} can be w_{dl} or w_{hl}, which are two different weighting factors for d_l and h_l respectively.
w_t	Weighting factor for trust.
E_l	Available energy of sensor node l.
T_l	Trust level of sensor node l.
N_l	Number of neighbouring sensor nodes.
W_l	Selection value of sensor node l in the process of new group leader selection.

In the new group leader selection process, the current group leader will send a selection process packet to specific nodes that are to participate in the selection process. All participant nodes will calculate their selection value W_l using the formula shown below, and send it to the current group leader:

$$W_l = (E_l \times w_e) + (T_l \times w_t) + (N_l \times w_n) + (\frac{1}{S_l} + w_{sl})$$

$$\text{If } E_l > \delta \text{ and } T_l > \alpha \text{ and } N_l > 0 \text{ and } S_l = d_l \text{ or } h_l > 0,$$

$$\text{otherwise } W_l = 0.$$

In this formula we multiply available energy E_l with weighting factor w_e, trust T_l with trust weighting factor w_t, N_l with its weighting factor w_n and S_l with its weighting factor w_{sl}, and then add them together to calculate W_l, in order to help us during simulation to evaluate the affect of each parameter during the selection process. Any node with W_l=0 will not be selected as a new group leader. In order to participate in the group leader selection, the current sensor node's available energy E_l, trust level T_l and number of neighbouring sensor nodes N_l need to be greater than the thresholds δ, α and zero respectively. We note that $1/S_l$ gives a higher value when a sensor node is closer to the current group leader.

The group leader sensor node will determine the highest value of W_l in order to select a new group leader sensor node. The current group leader will then broadcast a message containing the new group leader ID to all its group member sensor nodes. Furthermore the current group leader will send the master key and any keys shared with other groups to the new group leader. For future communication all data should then be routed toward the new group leader sensor node.

In the given formula we calculate the selection value W_l using four different factors by adding the values of the *energy* factor, *trust* factor, *number of neighbouring nodes* factor and *distance between group leader and node* factor. If node l has the highest energy

above the threshold δ, more than one neighbouring nodes ($N_l > 0$) and is close to the current group leader (S_l) but the trust T_l is less then the threshold α, then the node will have $W_l = 0$ and be automatically ruled out as a future group leader. Similarly the other factors will play roles in much the same way.

According to the results described earlier in Section 5.1.2, we don't involve all group member sensor nodes in the selection process. As nodes closer to the current group leader are more ideal for performing the role of the new group leader as compared to sensor nodes that are further away. Therefore we only involve the limited number of sensor nodes around the current group leader in the selection process. This will help us to reduce the energy cost during the selection process. This is in contrast with existing proposed solutions, which involve all group members in their election or selection process, thereby consuming extra energy as a result. This is another advantage of our proposed scheme. The formula above can be coded into all group member nodes. Whenever a group leader requests a member node's selection value, the node will calculate it using this formula and send it to the group leader. We will describe in detail the implementation of this group leader selection process in Chapter 6.

Moreover our current solution is also scalable. The number of participant nodes can be increased by sending participation packets to more nodes during the new group leader selection process. For example, the current group leader can only send participation packets to first hop neighbours or to first and second hops neighbours and so on.

In our present solution we do not consider issues when member sensor nodes send out false information (e.g. using incorrect trust levels for selection value calculations) in order to be selected as a new group leader. This is an area of ongoing research which we hope to tackle in future work.

5.2.3 Key Management for MSNs (Layer four)

In the previous chapter we briefly described our proposed key management solution for MSNs. In this section we will explain in detail about the solutions. Before going into further details we first present the security policy that we apply to MSNs.

5.2.3.1 Security policy for MSNs

We can establish different security policies for mobile sensor nodes when they roam from one group to another. After successful authentication, a new incoming mobile sensor node can only access limited services in the host group, for example to communicate with only a single host group member sensor node. Moreover the guest mobile sensor node might have many neighbouring sensor nodes within its communication range but according to the security policy, a guest sensor node is only entitled to communicate with one host sensor node. Furthermore the new mobile sensor node will not be involved in any internal operations of the group such as data aggregation, group leader selection, etc. The initial service available to the guest mobile sensor node will be routing.

The host group will establish future relations (similar to granting a membership) with the guest mobile sensor node on the basis of its past performance and behaviour history inside the group. Once the guest mobile sensor node achieves a membership of the group, it can then establish communication links with more than one sensor node, and participate in sensing and new group leader selection operations. These policies can be defined according to the target application nature and its security requirements.

The future stay of mobile guest sensor nodes in the host group will be monitored on the basis of a trust point system. If a guest sensor node performs a malicious activity it will be marked down with a negative trust point. The host group will make a decision based on these trust points as to whether to refuse or extend a further stay of the guest sensor node within the group.

In summary a group leader sensor node will allow a guest mobile sensor node to establish a secure communication link (through the sharing of a session key) with only a single sensor node, i.e. a new incoming sensor node will have only one neighbouring sensor node. The new sensor node can then only communicate through this one sensor node. We enforce such restrictions according to the threat level of the target application. Furthermore a group leader will grant more rights to incoming sensor nodes to establish secure communication links according to their trust levels.

5.2.3.2 Keypool based authentication for MSNs

In this solution we extend SADI-GKM for key establishment with the addition of a probabilistic key based idea for mobile sensor node authentication. This extended SADI-GKM pre-distribution phase includes the following activities:

1. Assign a unique ID to each sensor node, which is a combination of its group ID, *GI*, and its own ID, *NI*.
2. Assign a unique master key M_{kGI} to the leader of group *GI* and its member sensor nodes.
3. Pre-load *k* (key ring) keys from key pool *S*, which the mobile and static sensor nodes use for intra and inter group authentication.
4. Assign a unique value $U_{a,b}$ to every pair of neighbouring group leader sensor nodes *a* and *b*.
5. Generate and store a unique key $K_{GI, IDi}$ using the master key M_{kGI} and node ID of each sensor node.
6. Assign a unique key $K_{SI,IDi}$ to every sensor node ID_i (note: this key need only be assigned if we require the aggregation of data in an encrypted form at group leader nodes).

As $K_{SI,IDi}$ is a unique encryption key for each sensor node in the WSN, therefore it will not change even after the mobile sensor node roams from one group to another. However we do need to update the authentication keys which will be tackled later in this section.

Note that we assume the initial group IDs for the nodes are known in advance, however once deployed the protocol allows nodes to move between groups unrestricted.

We have amended the step 4 of the original SADI-GKM pre-distribution phase as shown above. According to the amended step 4, the *k* keys will be preloaded from the key pool *S* into every sensor node before deployment, whereas in the original SADI-GKM every

sensor node in each group *GI* is preloaded with a value V_{GI} for authentication inside the group and every group GI_i has a different V_{GIi}.

We are interested to find the probability of key sharing when a mobile sensor node moves from one group to another. We need to use a probabilistic key establishment idea in the case when a mobile sensor node moves from one group to another. What will be the probability that it can establish at least one link with any member of the new group? When a mobile sensor node moves from one place to another, we calculate the probability *P* of any specific node within the group being within radio range. We need to find the probability that the mobile sensor node can establish at least one link with one of its neighbouring sensor nodes. We do this by assuming the sensor nodes lie on a plane and that the radio coverage of a sensor node is circular with radius *r*. The area of radio coverage for the sensor node is therefore the area of a circle $\pi \times r^2$. If the overall area of the group is represented by *A* then the proportion of the group within the radio range of a single node can therefore be represented by *D* where

$$D = \frac{\pi \times r^2}{A}.$$

Here *r* is the radio range of the mobile sensor node in metres, *A* is the area of the new group, measured in m^2 and the constant π represents the ratio of the circumference of a circle to its diameter.

Having established *D* we need to find the probability that this sensor node shares a key with one of its neighbours. Furthermore we have already described in our security policy that initially a guest sensor node will be allowed to communicate with only one sensor node, depending on the given security requirements.

Given *D* as above, group size *N* and probability *P* that two given nodes do not share a key, the probability that the mobile sensor node does not share a key with any of its neighbours can be calculated as follows.

$$P_r = [(1 - D) + (D \times P)]^N. \quad (1)$$

The formula (1) includes the chance of a given node falling outside the radio range of the mobile node is $(1 - D)$. The chance that it is within the radio range but shares no key with the mobile node is $(D \times P)$. The chance that it is not possible to communicate with the node for either reason is therefore the sum of these two probabilities. Finally the probability P_r for N nodes is therefore this probability powered to the N-th degree.

Work on existing probabilistic key management solutions (key pool based ideas) has attempted to find the probability of establishing a secure link between a pair of sensor nodes. For example, according to the Basic Probabilistic scheme [72], if the size of the key pool S is 10,000 and 75 keys ($k = 75$, key ring) are preloaded in every sensor node, then the probability of sharing at least one key between a pair of nodes is 0.5. They formulated the probability that two sensor nodes do not share any keys as:

$$P = \frac{((S-k)!)^2}{(S-2k)!S!}. \quad (2)$$

By simplifying the expression of equation (2) we get:

$$P = \frac{((1-\frac{k}{S})!)^{2(S-k+\frac{1}{2})}}{(1-\frac{2k}{S})^{(S-2k+\frac{1}{2})}} \quad . \quad (3)$$

By substituting equation (3) into equation (1) we get:

$$P_r = [(1 - D) + (D \times \frac{((1-\frac{k}{S})!)^{2(S-k+\frac{1}{2})}}{(1-\frac{2k}{S})^{(S-2k+\frac{1}{2})}})]^N. \quad (4)$$

The probability of the mobile sensor node being able to establish one link out of the $D \times N$ neighbouring sensors is

$$P_r' = 1 - P_r.$$

Substituting the values from equation (4) we get

$$P_r' = 1 - [(1 - D) + (D \times \frac{((1-\frac{k}{S})!)^{2(S-k+\frac{1}{2})}}{(1-\frac{2k}{S})^{(S-2k+\frac{1}{2})}})]^N \quad . \quad (5)$$

Equation (5) will give the probability for the mobile sensor node to establish at least one link (enabling it to authenticate itself), given that we know the group area A and size N, mobile node radio range r, key pool size S and key ring size k.

The main purpose of finding the probability for a mobile node to share a key in the host group is authentication. Therefore the probability needs to be sufficiently high for the mobile node to authenticate itself to the new group. This formula also helps to find the probabilities of mobile sensor node authentication at different positions in the host group which we will revisit in section 6.3.9.

5.3 Summary

In this chapter we have described in detail all layers of our protocol stack. We began by describing our pre-design investigations. These investigations are important part of our research, which resulted in the identification of some core information needed to help us towards the development of an efficient protocol. In the detailed design section we described each of the four layers. For the first and second layers, *key management* and *secure data aggregation,* we described the basic key management and secure data algorithms together. Then we explained the third layer, *secure group leader selection,* where we have presented our novel formula for new secure group leader selection. In the

fourth layer we presented a key management solution for MSNs and an authentication model for mobile sensor nodes.

Chapter Six: SADI-GKM Implementation and Evaluation

6.1 Introduction

The previous chapter described the model of our novel SADI-GKM protocol for WSNs. In this chapter we present the evaluation and implementation of our work. As our work is divided into four different layers, we have therefore evaluated the performance of each of these layers separately and compared them with existing proposed schemes from the literature. We describe the performance evaluation process and discuss the different simulation scenarios used to show the performance of each layer of our protocol. To begin, we have implemented the first layer of SADI-GKM and tested its resilience against node capture attacks using different topologies (tree, grid and random mesh). The purpose of this is to determine the level of topology independence by simulating node capture attack using various topologies. We also aim to show that it is more resilient against node capture and replication attacks than existing schemes by simulating node captures and detecting the fraction of total communication comprised for different schemes in WSN. Further we have implemented the second layer (secure data aggregation) to establish the impact on energy use of various encryption schemes during secure data aggregation. We simulate each encryption technique and run them over a number of cycles to compare energy use. For the third layer (secure group leader selection) we aim to show the group leader selection scheme is efficient by simulating the process with a variety of factors to establish optimum values such as selection frequency. Finally the fourth layer (key management for MSNs) of our proposed protocol extends the protocol for MSNs. We test the applicability of key management in mobile environment.

6.2 Implementation Phases and Simulation Framework

We have written our simulation framework and undertaken our project implementation in Java (JDK.1.6). As described earlier our protocol contains four different layers, each performing a different function. Each layer has been implemented as a distinct program with links to the programs of the other layers. The first step of our implementation was topology creation, since we needed to test our proposed solution on different topologies. For this we have implemented tree, grid and random mesh topologies. Furthermore we have implemented an adaptive routing algorithm on the top of these topologies. The advantage of using Java is that it has its own built-in security packages for encryption and decryption. We will explain in detail about the topology, routing and security packages in later sub-sections.

Phase 1: Topology Creation

(a) Tree Topology Setup

As discussed in section 5.2.1, we organise the entire WSN into groups of sensor nodes. Each group has 100 sensor nodes and is structured as a tree topology. Each sensor node is connected to its parent and between one and five child sensor nodes. The number of child sensor nodes for each sensor node is selected randomly. Figure 6-1 shows 100 sensor nodes organised in such a tree topology. Algorithm 6-1 describes the tree construction process. In the first step we select the total number of sensor nodes. In the second step the tree construction process selects a single node as the parent node for the entire group; then a random function will select the number of its child sensor nodes (between a minimum of one and a maximum of five). Further every child node will be selected as a parent node to select its child nodes. This process runs repeatedly until all sensor nodes have been assigned children. Moreover for the simulation process we assume every child node shares a key with its parent node.

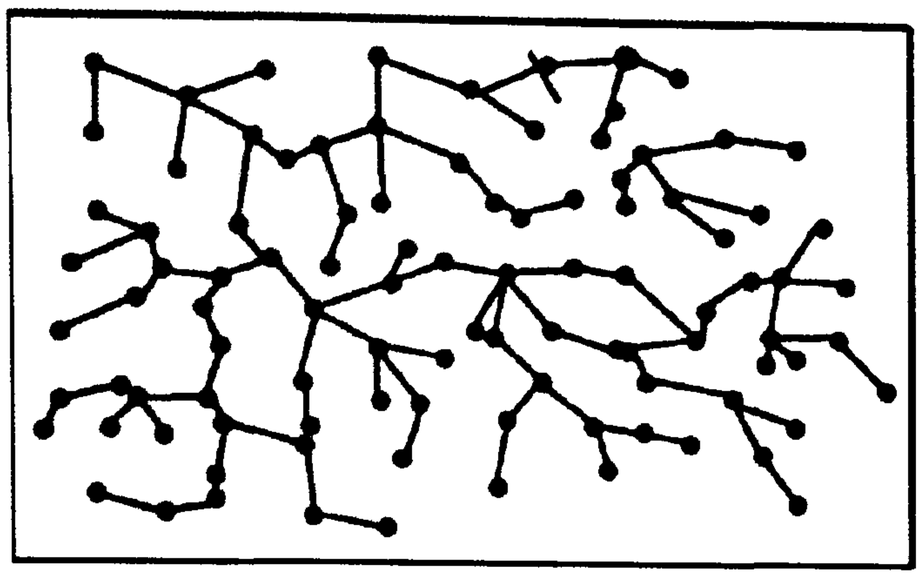


Figure 6-1: 100 sensor nodes in a random tree topology.

```
int u=0;
for(int i=0;i<tn;i++)  { // tn is number of total nodes in group
   if(totalnodes>0) { // if all nodes are assigned as child nodes
   n[i].parent=p; // assigning parent node
   r=no.nextInt(max)+1;
   n[i].left=node_c+1; // assigning left most child
   for(int j=0;j<r;j++) { // assigning child nodes
      //child assign
      if(tp<total_n) {  // Change if you want to change number of Childs
         tp=++node_c; // tp is temp variable
         n[i].child[u]=tp;
         if(r==j+1) //Assign Slib of every node
            n[i].slib[u]=0;
         else
            n[i].slib[u]=tp+1;
      }
      u++;
   }
   n[i].right=node_c; // assigning right most child
   totalnodes-=r;
   u=0;
   p++;
   }
}
```

Algorithm 6-1: Random tree topology construction with maximum five child nodes.

(b) Grid Topology Setup

This process is used to organise the entire WSN in a grid topology formation. We follow a similar process to that of the tree topology, managing the entire WSN in groups of 100 sensor nodes. In a grid of 100 sensor nodes every sensor node has a maximum of four and a minimum of two neighbouring sensor nodes, as shown in Figure 6-2. The java code for constructing this grid topology and assigning keys to relative sensor nodes are presented in Algorithm 6-2.

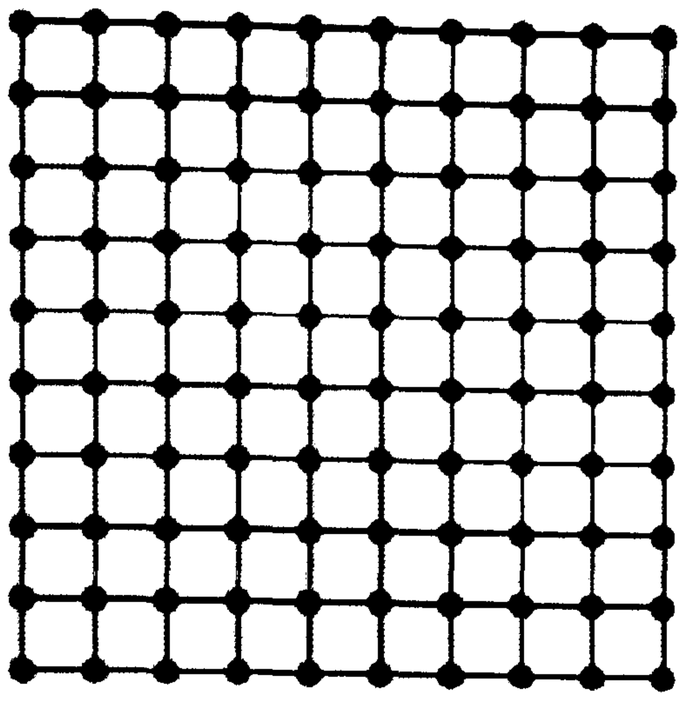


Figure 6-2: 100 sensor nodes in a grid topology.

```
Random n=new Random();
for(int u=0;u<100;u++) { // simulation run times
   //
   for(r=0;r<Max;r++) {
      for(c=0;c<Max;c++) {
            if ((r==Min && c==Min) || (r==Min && c==Max-1) || (r==Max-1 && c==Min)
                || (r==Max-1 && c==Max-1) ) {
                a[r][c]=2; check=1;
            }
        else if ((r<Max-1 && (c==Min || c==Max-1))
                || (c<Max-1 && (r==Min || r==Max-1))) {
                a[r][c]=3; check=2;
            }
        else {
                a[r][c]=4; check=3;
            }
         }
   }
}
```

Algorithm 6-2: Grid topology construction with maximum four and minimum two neighbouring nodes.

(c) Random Mesh Topology Setup

The setup of the random mesh topology is different from those of the tree and grid topologies. In a random mesh topology we select neighbouring sensor nodes on the basis of radio ranges between sensor nodes. Effectively, all sensor nodes coming within the radio range of a sensor node will form its neighbours. Sensor nodes in every group are randomly deployed and assigned geographical coordinates (x, y) as shown in Algorithm 6-3. We assume that the radio range of every sensor node is 40 m. The geographical area of each group is 150×150 m^2 containing 100 sensor nodes. Every sensor node in a group will establish communication links with its neighbouring sensor nodes falling in the range

of 40 m. Figure 6-3 shows the random mesh topology in a group. The pseudo code for constructing this random mesh topology is presented in Algorithm 6-3.

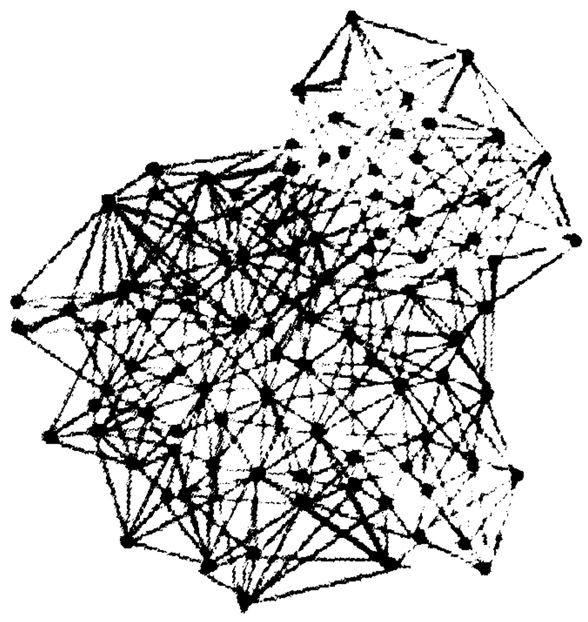


Figure 6-3: 100 sensor nodes in a random mesh topology.

```
for(int i=0;i<Max;i++) { // Initializing sensor nodes
     ne[i]=new Node();
}

Random n=new Random();
int x=0, y=0, check=0, ct=0;
for(int i=0;i<Max*2;i++) { // Deploy nodes randomly: assign random coordinates
   x=n.nextInt(Field); // Position of sensor nodes
   y=n.nextInt(Field);
   for(int j=0;j<Max;j++)
      if (row[j]==x && col[j]==y) {
         check=1; j+=100; // Break this loop
         }
   if(check==0) {
      if(ct<Max) {
         row[ct]=x; col[ct]=y; ct++;
      }
         else i+=Max*3;
     }
     else check=0;
}
for(int i=0;i<Max;i++) { tempr[i]=row[i]; tempc[i]=col[i]; }
for(int s=0;s<500;s++) { // Simulations runs
   //Reassign the nodes to same positions
     for(int i=0;i<Max;i++) {
         row[i]=tempr[i]; col[i]=tempc[i];
     }
     // Establishing the links with neighbouring nodes
   for(int i=0;i<Max;i++) {
      rr=row[i]; cc=col[i];
      for(int j=0;j<Max;j++) {
         r=row[j]; c=col[j];
         if (!(r==rr && c==cc)) {
                  if ((r>=rr-range && c>=cc) && (r>=rr-range && c<=cc+range)
                       && (r<rr && c>cc) && (r<=rr && c<=cc+range)) {
               count++; ne[i].r[j]=row[j]; ne[i].c[j]=col[j];
            }
                  else if ((r>=rr-range && c<=cc) && (r>=rr-range && c>=cc-range) &&
                  (r<=rr && c>=cc-range) && (r<=rr && c<=cc)) {
               count++; ne[i].r[j]=row[j]; ne[i].c[j]=col[j];
            }
                  else if ((r>=rr && c>=cc-range) && (r<=rr+range && c<=cc)
                       && (r<=rr+range && c<=cc) && (r>rr && c<=cc)) {
                  count++; ne[i].r[j]=row[j]; ne[i].c[j]=col[j];
            }
                  else if ((r>=rr && c>cc) && (r<=rr+range && c>=cc) && (r<=rr+range
                       && c<=cc+range) && (r<=rr+range && c<=cc+range)) {
               count++; ne[i].r[j]=row[j]; ne[i].c[j]=col[j];
            }
             } // closing if used to stop duplication
      } //closing loop
      node[i]=count; count=0;
   } //closing for loop
} //closing for loop
```

Algorithm 6-3: Random mesh topology construction.

Phase 2: Routing Algorithms

We have used the adaptive routing given in [166, 162] during our implementation. This routing algorithm will find a suitable neighbour, which will help to route data towards the sink.

We have chosen this adaptive routing algorithm due to ever changing traffic in our simulated networks. We have assigned a Directional Value (DV) to each sensor node. This Directional Value helps to route data towards the sink or a group leader sensor node, and helps the sink or group leader to find the location of a source sensor node. In all cases the final destination will be the sink or a group leader sensor node, so each sensor node knows in which direction to send its packets. Using our routing algorithm a sensor node can compute a suitable direction towards the sink or its group leader sensor node. The best route depends on the minimum Directional Value and the maximum energy available. As global routing has a huge amount of overhead and cannot be practically applied to WSNs [166], it is appropriate to use a local routing. The method to be presented below makes use of local information for routing data, including the remaining power of a sensor node, the number of its neighbours, the number of hops between the source and destination nodes, and the sum of the remaining power of the next neighbours.

The routing algorithm given below routes the data from different directions towards the sink or a group leader sensor node:

So_r Source row
So_c Source column
De_r Destination row
De_c Destination column
Si_r Sink row
Si_c Sink column

If $De_r = Si_r$ AND $De_c = Si_c$ then
    This is the Sink or Group leader
Else
    If $|So_r – Si_r| > |De_r – Si_r|$ OR $|So_c – Si_c| > |De_c – Si_c|$ then
        Send data in this direction
    Else
        Stop data in this direction
    End if
End if

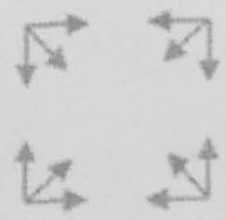


Figure 6-4: All of the possible directions the algorithm can use to route data.

We have tested the above algorithm with the settings shown in Figures 6-5. It should be noted that in the algorithm, a destination node refers to a source node neighbour, which may not be the sink or group leader sensor node. We use Figure 6-5 as an example to explain how the algorithm works. In this example, we use the notation (*group leader ID, source node x, source node y*) to represent a position of a sensor node in the topology. Suppose that sensor node (1,1,1) wants to send data to group leader (1,3,3). Now the source sensor node (1,1,1) can either select sensor node (1,1,2) or (1,2,1) as a destination node because both nodes can forward the data to the group leader. If node (1,1,2) is selected, it will receive the data from (1,1,1) and then become a source sensor node to forward the received data to another node. Node (1,1,2) has three neighbouring sensor nodes but can only select node (1,1,3) or (1,2,2) for sending the data forward because these two neighbouring sensor nodes are leading towards or closer to the group leader sensor node. Suppose that (1,1,3) is selected as a destination. When (1,1,3) receives the data, it will then act as a source. Similarly (1,1,3) has three neighbours but only two of them, (1,1,4) and (1,2,3), could lead towards the group leader. At this point the algorithm will select node (1,2,3) as a destination instead of (1,1,4), because (1,2,3) is closer to the group leader than (1,1,4). Similarly node (1,2,3) becomes a source and will send the received data to node(1,3,3) which in this case is the group leader sensor node.

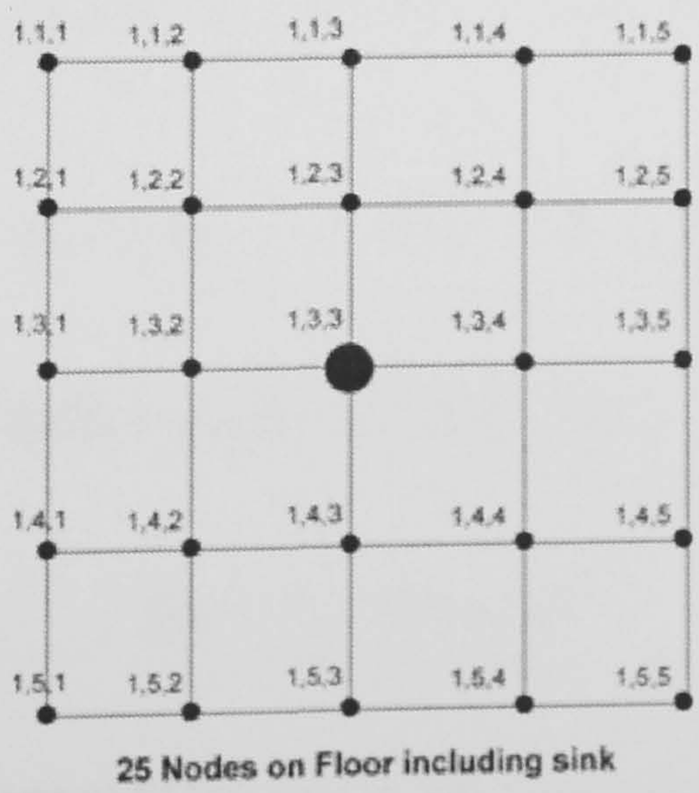


Figure 6-5: 25 nodes arranged in a grid including a group leader [162].

Phase 3: Radio Model

We use the same radio model as LEACH [3-28], which is widely used by many researchers [167 – 171]. Currently, there is a great deal of research in the area of low-energy radios. Different assumptions about radio characteristics, including energy dissipation in transmit and receive modes, will change the advantages of different protocols. In our work, we assume a simple model where the radio dissipates E_{elec} = 50 nJ/bit to run the transmitter or receiver circuitry and ε_{amp} = 100 pJ/bit/m^2 for the transmit amplifier to achieve an acceptable E_b/N_o (see Figure 6-6 and Table 6-1). We also assume a d^2 energy loss due to channel transmission (where d is the distance of transmission). Thus, to transmit a k-bit message with a distance d, the radio would expend the following energy based on this radio model [139, 167 – 171]:

$$E_{Tx}(k, d) = E_{Tx\text{-}elec}(k) + E_{Tx\text{-}amp}(k, d)$$

In other words, the energy required is the sum of the energy needed to run the transmitter circuitry for k bits and the energy needed for the transmit amplifier to allow the k-bit message to be sent a distance d. Since the transmitter circuitry requirements remain constant per bit, and the amplifier requirements are proportional to the square of the distance to be transmitted, we can see that this is equivalent to the following energy requirements.

$$E_{Tx}(k, d) = (E_{elec} \times k) + (\varepsilon_{amp} \times k \times d^2) \quad (1)$$

To receive this message, the radio expends the following:

$$E_{Rx} = E_{Rx\text{-}elec}(k)$$

In this case the energy requirements are just those of the receiver circuitry, which remain constant per bit received. Hence we can see that this is equivalent to the following energy requirements.

$$E_{Rx}(k) = E_{elec} \times k \quad (2)$$

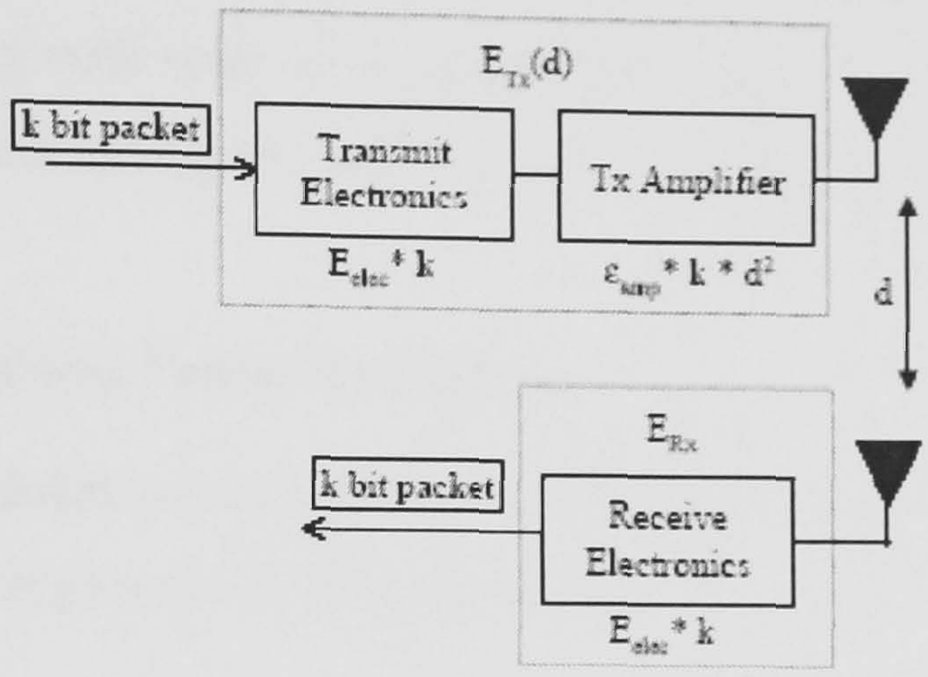


Figure 6-6: First order radio model [139].

E_{Tx} (k, d) is required in order to calculate the cost of transmitting the packets over distance d and $E_{Rx}(k)$ value is important in order to get the receiving cost of a packet.

Operation	Energy Dissipated
Transmitter Electronics ($E_{Tx\text{-}elec}$) Receiver Electronics ($E_{Rx\text{-}elec}$) ($E_{Tx\text{-}elec} = E_{Rx\text{-}elec} = E_{elec}$)	50 nJ/bit
Transmit Amplifier (ε_{amp})	100 pJ/bit/m^2

Table 6-1: Radio characteristics [139].

Phase 4: Security Implementation Library

We have used the java security library for our key management protocol implementation. During implementation the important functionalities needed were to generate keys, and to encrypt and decrypt data using these keys. We have generated the keys using hash. Furthermore we used two different methods for data encryption and decryption: blowfish

and homomorphic cryptography. The process for homomorphic encryption was explained earlier in chapter three. Every sensor node in the network has three main cryptography-related functions along with their other capabilities: `generate_key()` to produce a new key, `encrypt()` for data encryption and `decrypt()` for data decryption.

Phase 5: Sensor Node and Packet Definitions

During the implementation we have defined a *Node* class which contains various methods and attributes. These implemented methods include:

- `Sense ()`: Used by every node for sensing its environment.
- `Neighbours ()`: Finds and stores information about its neighbouring sensor nodes.
- `Send ()`: Sends data packets to a destination node.
- `Receive ()`: Receives data packets from a sender node.
- `Find_route ()`: Finds an energy efficient route, which implements our routing algorithm.
- `Aggregate ()`: Used by group leader nodes to aggregate collected data.
- `Generate_key ()`: Used by sensor nodes to produce new keys.
- `Encrypt ()`: Used for data encryption.
- `Decrypt ()`: Used for data decryption.

Along with these functions there are a number of attributes, including *energy*, *trust*, *number of hops from group leader node*, and *number of neighbouring nodes.*

Furthermore we have also defined a *Packet* class. Each instance of the Packet class contains data and its destination address. The packet size can be different for different applications according to the nature and size of the sensed data. For example in an application where sensor nodes need to sense the temperature of the environment, a 1 byte (8 bit) data may be enough. However this one byte represents the data in its unencrypted form. Therefore after encryption the data packet size may increase. In our

case, every packet contains encrypted data, a hash value (of the encrypted data, authentication value, and sender node ID and time stamp), the sender sensor node ID and its time stamp. The size of the encrypted data depends on the encryption algorithm used, such as blowfish or DES.

Phase 6: Network and Security Assumptions

We assume a static and synchronized WSN initially. The sink, acting as a key server knowing all the groups' master keys, is assumed to be a PC, laptop or computer with indefinite power capabilities. Each group leader sensor node acts as an aggregator and router sensor node for communication with other groups. Sensor nodes can be deployed via aerial scattering in outdoor applications. However in indoor applications sensor nodes can be installed manually and their immediate neighbouring sensor nodes will be known in advance according to the application requirements.

We assume an adversary can eavesdrop on all traffic, inject packets, or replay older messages. If a sensor node is physically compromised, all the information it holds will be known to the attacker. However, the base station or sink cannot be compromised.

6.3 Performance Evaluation and Simulation

In this section, we analyze our proposed scheme in detail. For analysis, we adopt similar methods to those described in Du *et al.* [78]. We evaluate our proposed scheme against the following criteria that represent desirable characteristics in a key distribution scheme for WSNs: stronger resilience against node capture, forward and backward secrecy, resilience against replication attacks, secure data aggregation, memory overhead, communication overhead and connectivity.

We have considered the energy consumption of data processing (encryption, decryption and aggregation), and that of sending and receiving packets. Further detail was given in section 6.2.

In the next section we focus mainly on node capture attacks and secure data aggregation, providing detailed information about our proposed protocol's resilience. In the evaluation

we make no assumptions about the topology used for communication inside a group, as our proposed solution is structure and density independent. As part of the comparison, we have simulated the SADI-GKM protocol using three different topologies: tree, grid, and random mesh.

6.3.1 Node Capture Attacks

In this section we check the performance of our SADI-GKM protocol against node capture attacks. DGKE has shown different resilience to node capture attacks for different topologies, highlighting its topology dependence [163]. However, in contrast to this we are able to show that our SADI-GKM Protocol generates similar results with different topologies to prove it is topology independent. This is because in SADI-GKM every sensor node uses a single key for encryption and does not share keys with any neighbouring node, therefore compromise of any single node in a network will not help an adversary to compromise the communication of other sensor nodes. SADI-GKM key sharing method does not depend on the structure of the topology. However in DGKE every sensor node used multiple keys for encryption according to the number of its neighbouring sensor nodes as described in section 5.1.1. Consequently DGKE behaves differently when used in different topologies. A summary of these results is shown in Figure 6-7, illustrating the relative structure independence of the protocol.

As a further performance analysis against node capture attacks we implemented our protocol using grid and mesh topologies and compare it against existing grid and mesh-based schemes in the following sections. Finally we present the performance of our proposed protocol in various different scenarios.

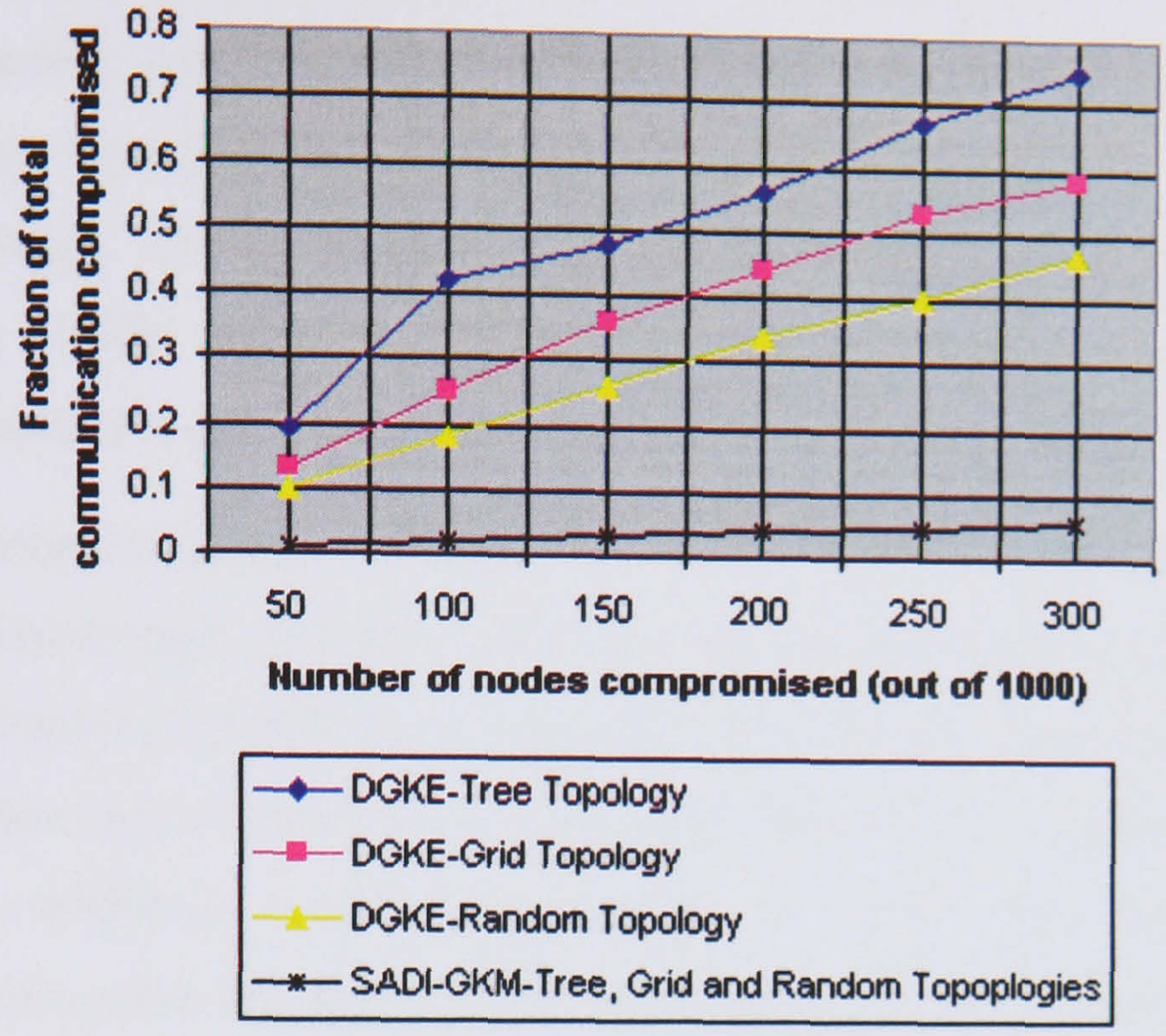


Figure 6-7: Effect of node capture attacks on DGKE and SADI-GKM for different topologies.

(a) SADI-GKM Protocol Performance Using Grid Topology

To evaluate the performance of our scheme using grid topologies, we compare our results with *PIKE 2D* [76] and DGKE grid-based key management protocols. We consider PIKE 2D since it is a well established and widely known protocol that uses a unit distance bidirectional communication model. Consequently, we were able to use the same simulation and WSN settings for our SADI-GKM protocol, PIKE 2D and DGKE. PIKE and DGKE were simulated on a flat, square deployment field. Although we note that there are other functional differences between the various protocols, our focus of concern is primarily on that of security and our objective is therefore to compare the protocols based on this metric (resilience against node capture attacks), rather than other aspects of their functionality.

We have used the following configuration during the simulations. The WSN is based on 5000 sensor nodes. The link density at each sensor node is two to four sensor nodes. We take the top right, top left, bottom right and bottom left edge sensor nodes to be group

leaders for different groups. During the simulation, if a group leader is compromised, we assume that the entire group communication is compromised as the group leader node holds a master key. Note that we didn't consider the second, third and fourth layers of our protocol at this stage. The simulations involved 50 groups of sensor nodes, with each group comprised of 100 sensor nodes. The results are based on averaged runs of 500 simulations. A detailed explanation of the simulation with code is given in [6].

The results illustrated in Figure 6-8 show that our SADI-GKM protocol provides better resilience against node capture attacks. Through a comparison of the three algorithms, we can see that this improved performance is primarily as a result of fewer keys being shared on average between sensor nodes in our proposed protocol as compared to DGKE and PIKE 2D. This reinforces the first observation of DGKE that sharing keys with neighbouring sensor nodes for data encryption and decryption helps node capture attacks to compromise the entire WSN.

For these results, the proportion of compromised communications is based on the number of compromised links as a fraction of the total links in the simulation.

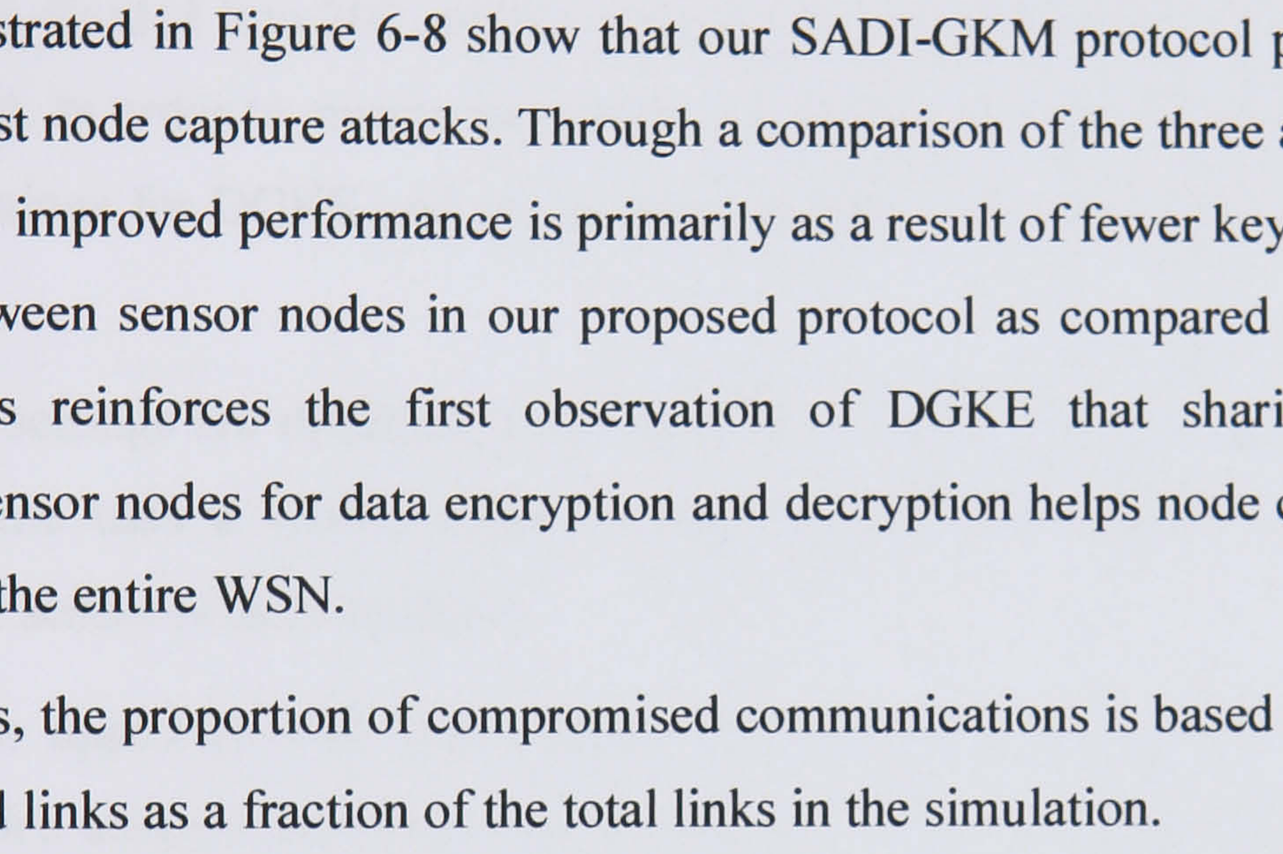


Figure 6-8: Comparison of probability of total communication compromise in grid topology.

(b) SADI-GKM Protocol Performance Using Random Mesh Topology

For the performance evaluation of the SADI-GKM protocol using a random topology (random mesh), we compared our results with the group-based key management protocol *Group-based EG* [74] and DGKE random mesh. We chose the Group-based EG scheme since the scheme organises the WSN into groups with each group being organised using the random mesh topology. The Group-based EG scheme assumes a key pool size of 10,000, which is divided into 200 smaller, equal-sized key pools with 500 keys for each smaller key pool. In order to ensure comparable results, we have used similar simulation and network settings for DGKE and our proposed protocol as were available for Group-based EG.

The simulation settings are different from the PIKE 2D and DGKE grid, as the Group-based EG scheme uses a 10,000 sensor nodes WSN for the simulation and the link density of every sensor node is random.

To compare our approach with Group-based EG and DGKE we have used a similar configuration. We assume that there are a total of 10,000 sensor nodes deployed in a 1000×1000 m^2 area. These sensor nodes are divided into 100 deployment groups with 100 sensor nodes in each group. The internal communication structure of every group will be distinct from other groups. We assume a radio range of $R = 40$m. Every sensor node will find its neighbouring sensor nodes within a 40 metre radio range, thereby dictating the link density. The results are based on averaged runs of 500 simulations. Figure 6-9 shows that the SADI-GKM protocol clearly has better resilience against node capture attacks in mesh topologies.

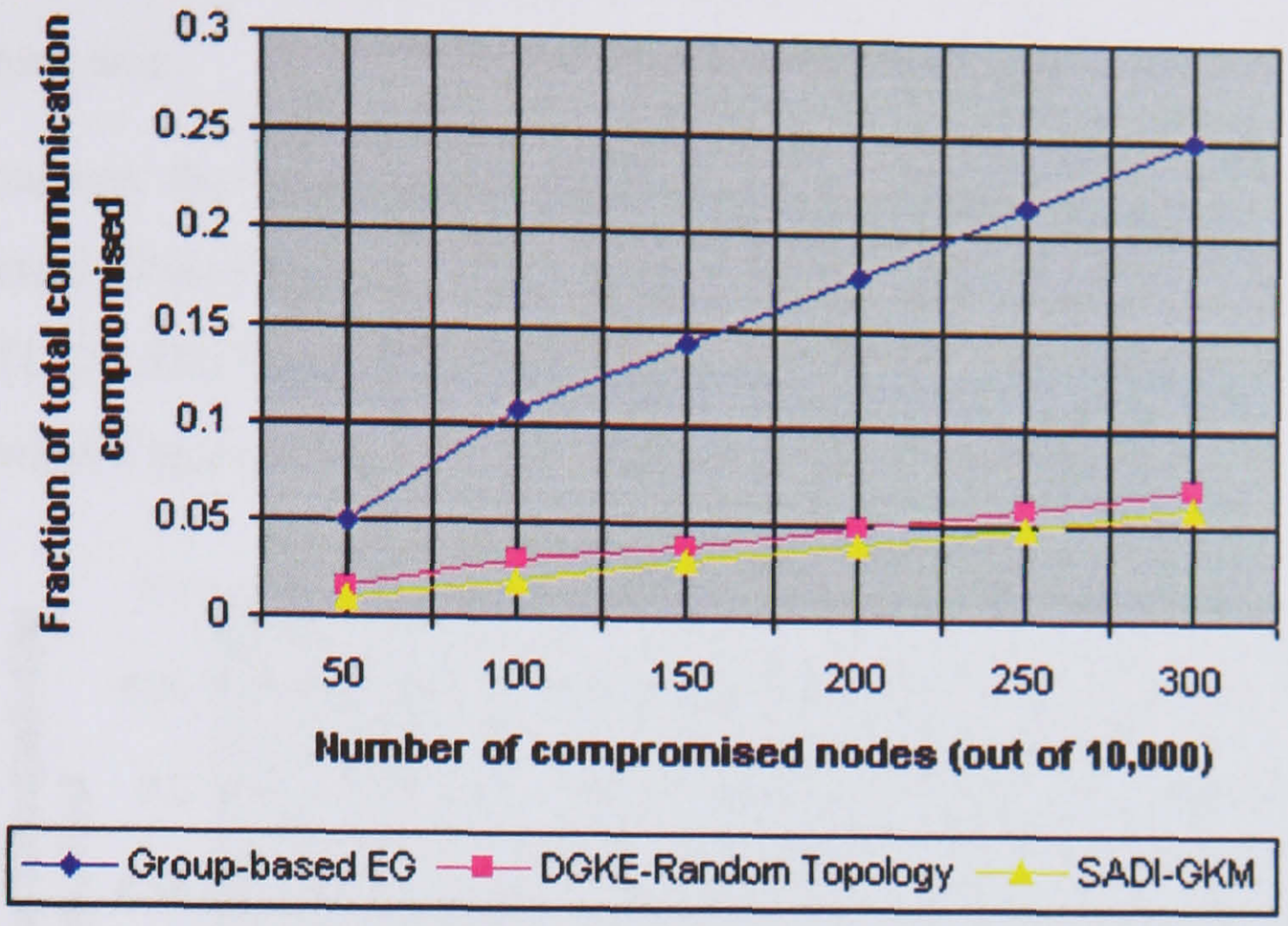


Figure 6-9: Comparison of probability of total communication compromise in random mesh topology.

Again, this is partly a result of the way keys are distributed between sensor nodes. As we have already noted, the Group-based EG scheme utilises a key pool in order to allow keys to be shared between sensor nodes. This has certain advantages in terms of memory usage, where a sensor node is unable to hold a unique key for every other sensor node in the WSN. It also marginally increases security during the WSN deployment stage, avoiding the need to use a master key for all sensor nodes in a group. However, the memory benefits for group-based systems are less pronounced, and the consequence of using a key pool is greater key sharing across multiple sensor nodes. This compares against our own scheme where key sharing is restricted to pairs of sensor nodes, providing the increased resilience shown by the simulation results.

We can see in Figure 6-8 that after compromising 250 sensor nodes 10% of communication is compromised in the network size of 5000 sensor nodes using grid topology. Furthermore in Figure 6-9 after compromising 250 sensor nodes 5% of communication is compromised in the network size of 10,000 senor nodes using random mesh topology. These results also prove SADI-GKM topology independency. The

difference in fraction of communication compromise in both figures is because of different network size.

Finally we compare the SADI-GKM protocol random mesh results with other similar existing schemes *Random Key Pool* (RKP Du et al.) [78] and *State-Based Key Management* [172]. The State-Based Key Management scheme improves on the results of others as shown in Figure 6-10.

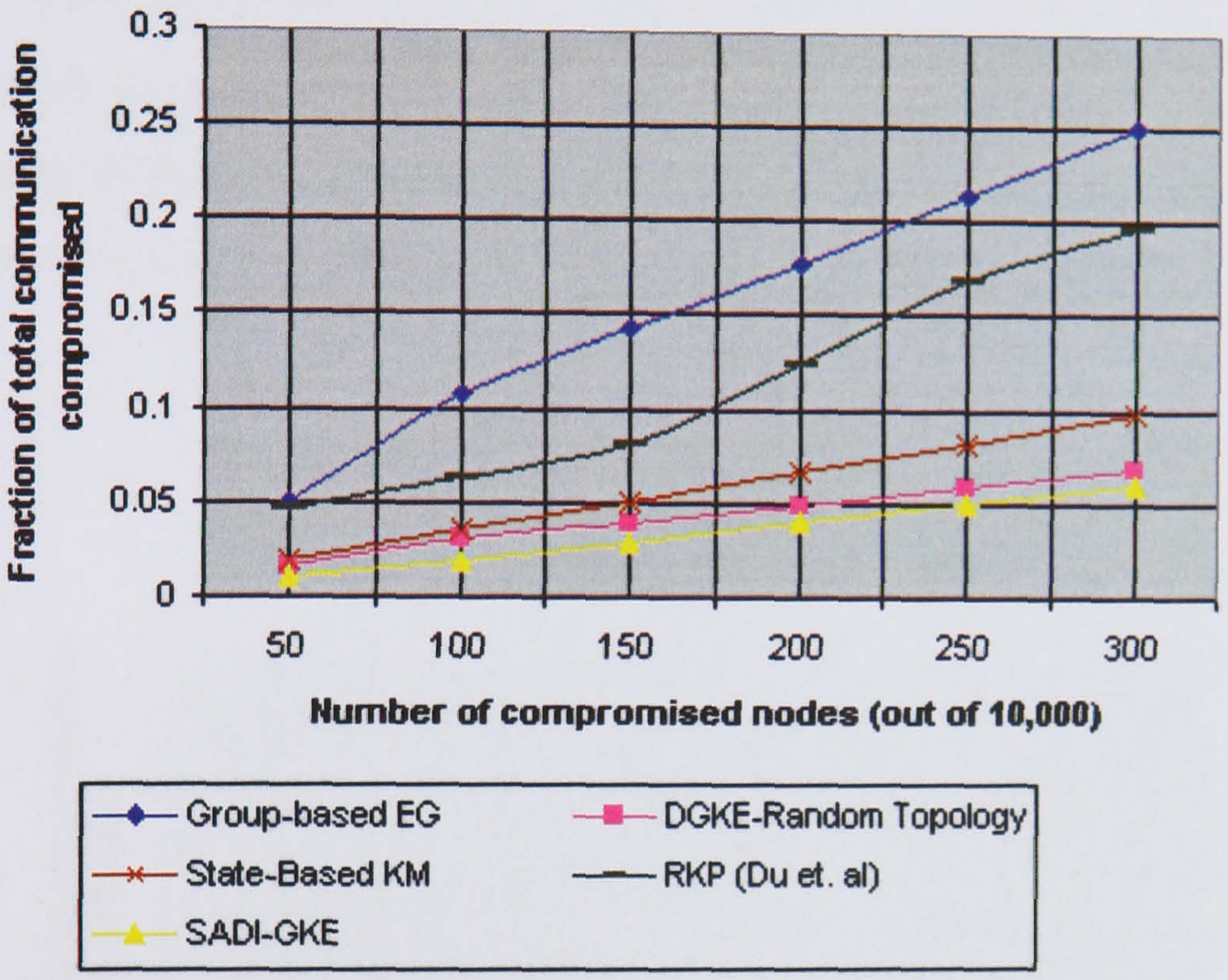


Figure 6-10: Comparison of probability of total communication compromise with other existing scheme.

Although our proposed solution is a group based solution, we have only considered peer to peer communication inside the groups. In contrast to our solution, the other schemes compared against provide the added functionality of secure peer to peer communication for the entire sensor network. However as we shall see in the following section this comes at a cost in terms of scalability, memory overhead and processing. In situations where such functionality is not needed, our solution therefore provides important benefits.

(c) Resilience Against Node Capture Attacks With or Without Using Groups

In this section we have undertaken an analysis as to whether WSNs are more robust against node capture attacks either with or without using groups. We have implemented the SADI-GKM protocol in both group-based and non-group-based WSNs using a random mesh topology. Figure 6-11 presents the effect of node capture attacks on WSNs both with and without groups using SADI-GKM, and the probabilities of group leader compromise using different topologies are shown in Figure 6-12.

We can see from Figure 6-12 that the probabilities of group leader compromise are similar across the different topologies. The effect of grouping therefore remains the same using different topologies in the case of the physical compromise of the group leader sensor nodes.

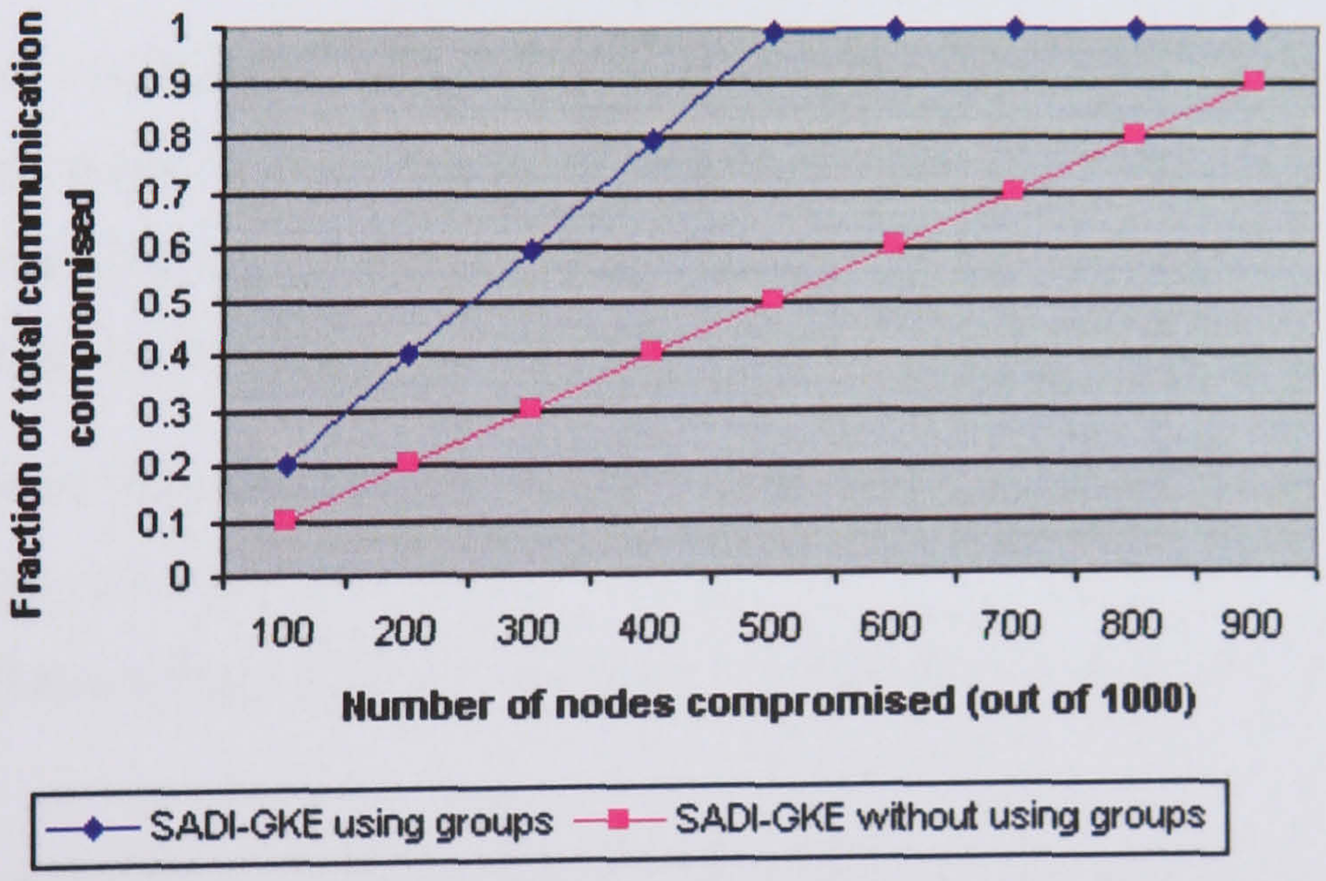


Figure 6-11: Node capture attacks in group and non-group SADI-GKM.

The results for the non-group based solution are better than those for the group-based solution. An important reason for using groups is that of scalability. Otherwise our proposed protocol produces better results without using groups. Therefore in small-scale WSNs we can use our proposed protocol with greater efficiency.

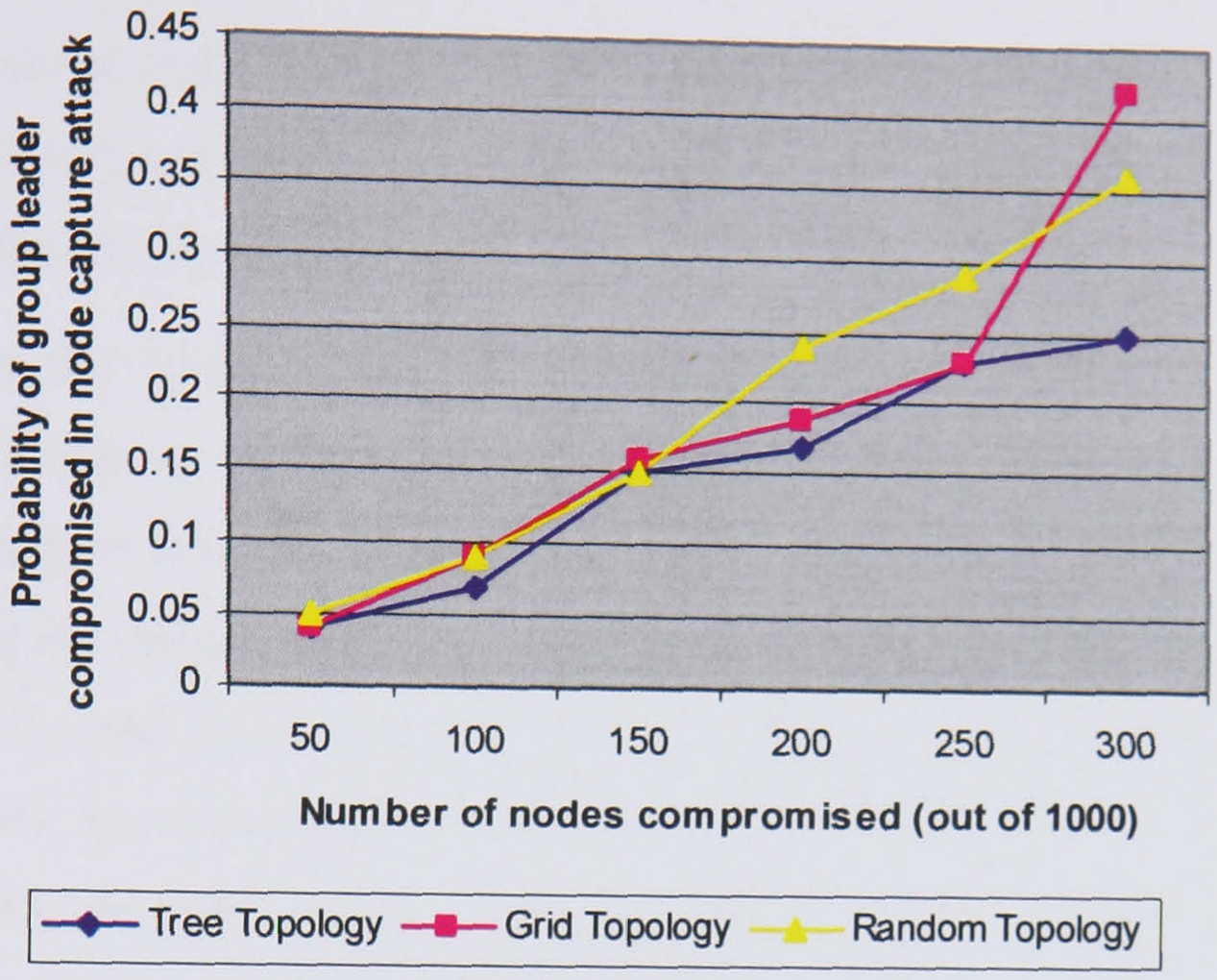


Figure 6-12: Probability of compromising group leader in different topologies.

In summary, our solution can easily be used in small and large-scale WSNs with high resilience against node capture attacks, compared to the other solutions that we have tested against. Our simulations show the influence of these three parameters, which were highlighted initially by DGKE:

- The structure (topology) of the WSN
- The density of the sensor nodes
- Sharing keys with neighbouring sensor nodes used for encryption and decryption.

In our proposed protocol we have avoided the influence of these three parameters, thereby obtaining significantly improved results in terms of resilience against node capture attacks, memory overhead, processing and connectivity.

6.3.2 Replication Attacks

To prevent replication attacks we have used time stamps at two steps in our algorithms. First, each source sensor node encrypts a time stamp with the collected sensed data, to

prevent replication attacks between the source sensor node and destination (the group leader sensor node or sink). Second, a sender sensor node sends a time stamp TS_{IDi} with the value X_i, so that the receiver sensor node can authenticate the sender node, protecting against replication attacks at each hop.

We will explain this process using an example. Suppose sensor node B is an enemy sensor node and acts as if it is a normal group member. Now sensor node A sends information to sensor node B. At the first step A encrypts its sensed data along with a time stamp using the unique key $K_{GI,IDi}$. At the second step the encrypted data M_i together with the values V_{GI} and ID_i as well as time stamp TS_{IDi} are hashed together and assigned to X_i. As we know, V_{GI} is shared between all member sensor nodes, and $K_{GI,IDi}$ is different for every sensor node in the group. Finally A sends X_i, M_i, ID_i and TS_i to B (the enemy sensor node). To launch a replication attack, B will act as A, using A's ID. In other words, B starts forwarding the same encrypted data M_i using ID_i from A but using a new time stamp. We assume that B has value V_{GI} and A's ID_i. Although this is sufficient to convince intermediate sensor nodes, the group leader can finally detect this attack by checking the data freshness, since B is unable to produce a fresh M_i without knowing A's encryption key shared only between A and its group leader.

Unfortunately there is an energy overhead that may result from the forwarding of replicated packets by intermediate sensor nodes that may allow the potential for a denial of service attack. Our intention has been to safeguard data confidentiality and integrity, while minimising the overhead of having to store multiple keys at each node. In particular, providing stronger authentication at each individual node would incur additional overheads without totally removing the potential for denial of service attacks. Nonetheless, while we don't consider it further here, we do acknowledge that providing stronger authentication at each step can reduce the impact and potential for such attacks.

Furthermore our proposed protocol has the capability to add more layers and integrate with the existing four layers. The new layers can have different functionalities, for example prevention of denial of service attacks.

6.3.3 Secure Data Aggregation and Communication Overhead

In this section we present the energy consumption during the implementation of SADI-GKM with adaptive routing for WSNs. The simulations involved groups of 100 sensor nodes using a grid topology with one group leader sensor node. The initial energy of all sensor nodes is set to 1 Joule. We assume that all sensor nodes in the group will continually sense and send information to the group leader and that the group leader will aggregate this data and send it toward the sink. Once all sensor nodes have successfully sent information to the group leader it will have completed one cycle (100 events). Our simulation results are based on 20 to 700 cycles in a group of sensor nodes. Note that during this analysis we have not considered the authentication cost at every hop. However we present results with authentication costs in the next section.

During evaluation we have considered a number of different cases in order to establish the energy requirements for various levels of security functionality. As defined by the SADI-GKM protocol, the general structure of the WSN for these various cases is shown in Figure 6-13.

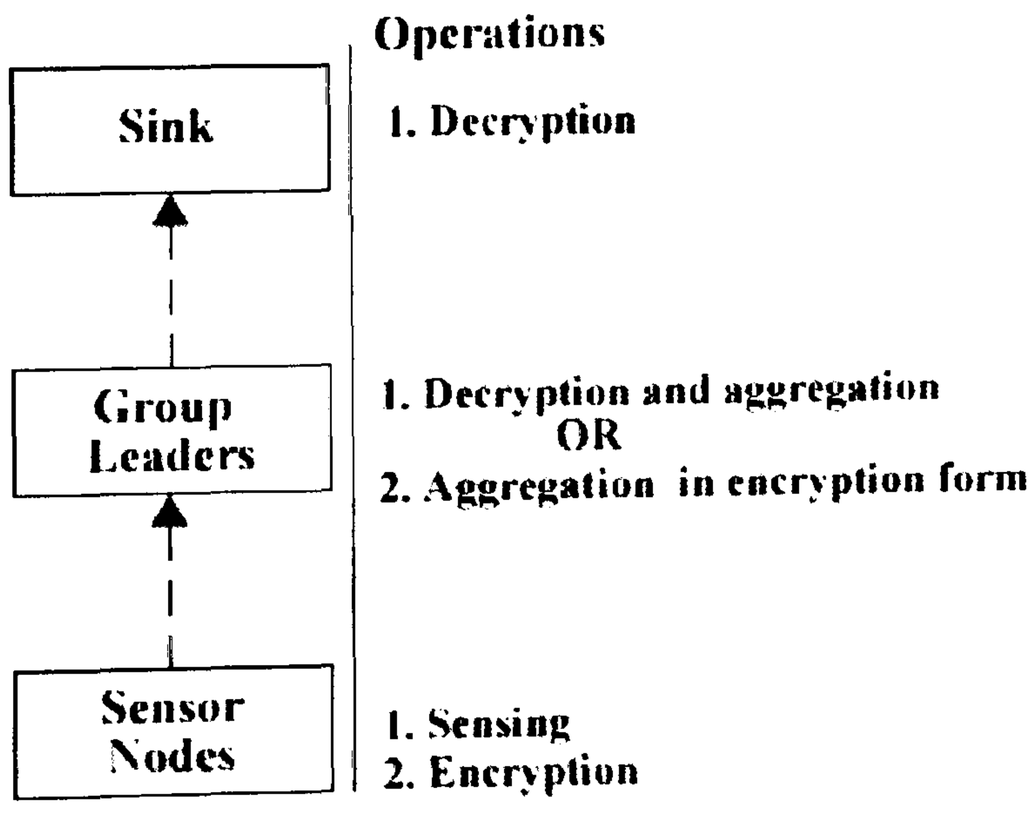


Figure 6-13: Overview of aggregation functionality provided in various cases by nodes in the SADI-GKM protocol.

SADI-GKM provides two different levels of confidentiality services (cases) at the group leader node according to the available resources and confidentiality requirements of different applications:

Case 1: Data can only be decrypted at the group leader sensor node, aggregated and re-encrypted using a master key before being sent to the sink.

Case 2: Data can be aggregated in an encrypted form without decryption and subsequently sent to the sink.

We have implemented Case 1 in three different forms. In the first form (*No-Security*) we remove all security features in order to establish the exact cost of SADI-GKM during communication. Therefore the cost of "No-Security" is only the cost of routing and aggregation at the group leader sensor node, as shown in Figure 6-14. The packet size in this first form for each event is 24 bits (e.g. 8 bits for sensed temperature data and 16 bits for source node ID_i).

In the second form every sensor node in the group encrypts its sensed data using the key $K_{GI,IDi}$ as described in Section 5.2. Furthermore the group leader sensor node will decrypt all received data and calculate the final aggregated result. The aggregated result will then be re-encrypted using the master key M_{kGI} and sent to the sink. In this second form we use the Blowfish algorithm for encryption and decryption which increases the size of the packet to 208 bits (192 bits of encrypted data and 16 bits of source node ID_i). The third form is similar to the second form but in this case we have used homomorphic encryption (as discussed in Chapter 3) instead of Blowfish for encrypting and decrypting the data. The simulation results for all these forms are shown in Figure 6-14. We can clearly see that using Blowfish encryption (since it takes a variable-length key, from 32 bits to 448 bits, making it ideal for a variety of uses), the group leader sensor node suffers complete energy loss after 440 cycles.

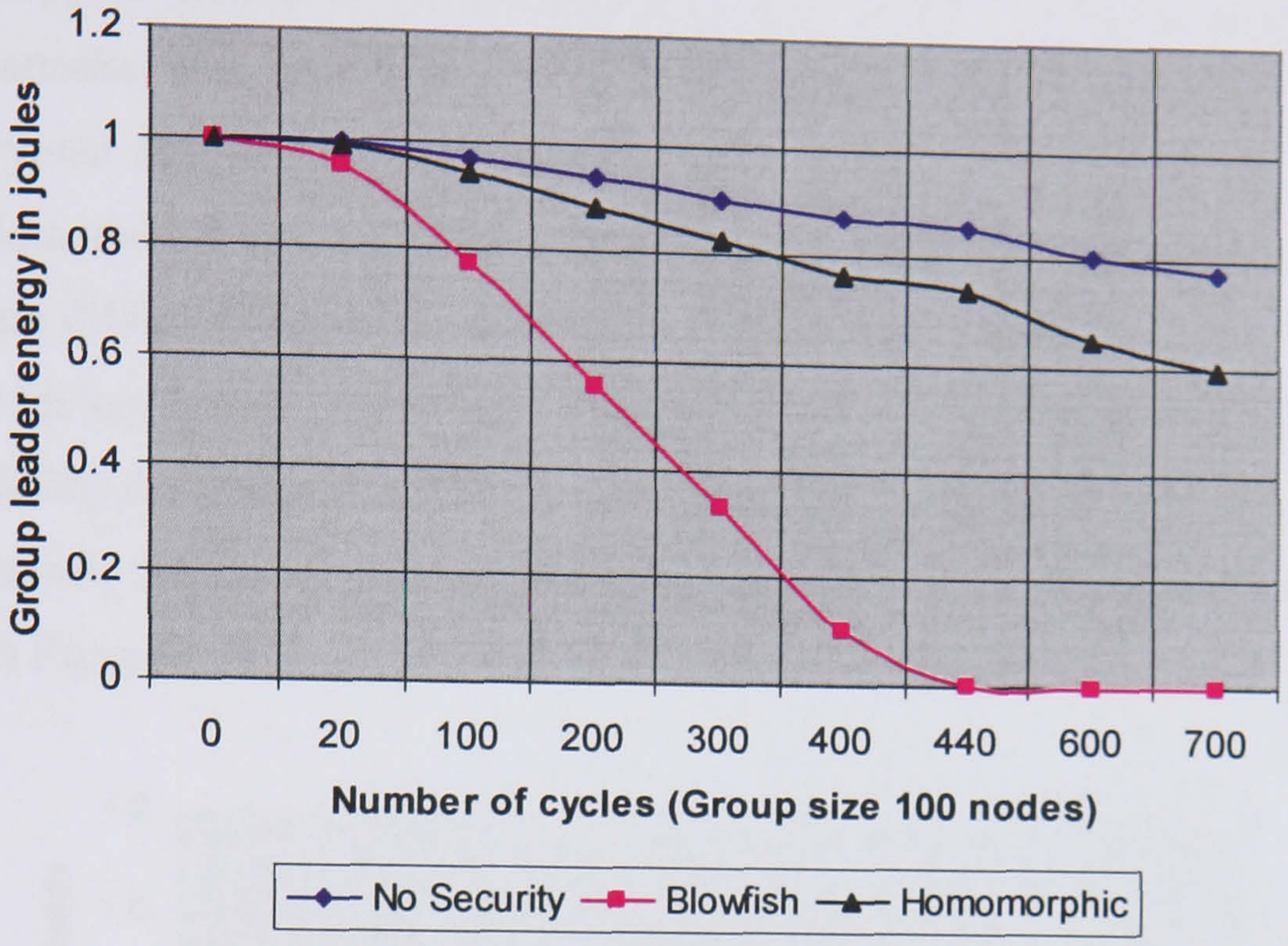


Figure 6-14: Cost of SADI-GKM using a group of 100 nodes in a grid formation.

Figure 6-14 clearly shows how the load on the group leader sensor node increases as we apply increased security (confidentiality) measures.

In Case 2 every sensor node ID_i has two keys: $K_{GI,IDi}$ shared with the group leader sensor node and $K_{SI,IDi}$ shared with the sink. In the first step a sensor node encrypts its data using key $K_{SI,IDi}$. In addition the encrypted data is re-encrypted along with a time stamp using $K_{GI,IDi}$. Consequently encryption occurs twice at the source sensor node. This additional encryption is used to provide resilience against replication attacks. Once the group leader sensor node receives a packet it will decrypt it using the master key M_{kGI} and the source node ID_i. Subsequently, the group leader aggregates the encrypted data, encrypts the aggregated result using master key M_{kGI}, and sends it to the sink.

We have also implemented Case 2 using the homomorphic and Blowfish encryption algorithms. However as described earlier, encryption happens twice at a source sensor node using the keys shared with the sink ($K_{SI,IDi}$) and group leader ($K_{GI,IDi}$). In the first form we use Blowfish twice for encryption with keys $K_{SI,IDi}$ and $K_{GI,IDi}$. In the second form we use homomorphic (using $K_{SI,IDi}$) and Blowfish (using $K_{GI,IDi}$) algorithms together. Blowfish is used for the second encryption, since it is necessary to encrypt the time stamp

with the encrypted data at the source sensor node in order to ensure resilience against replication attacks. The packet sizes used for the two forms are 208 bits (192 bits of double encrypted data and 16 bits of source node ID_i) and 400 bits (384 bits of double encrypted data and 16 bits of source node ID_i) respectively. The packet sizes for these two forms are different because in the latter we have applied Blowfish twice. As we are using a 160 bit key length, the Blowfish encrypted data size will always be a multiple of 192 bits. Adding the timestamp to the initially encrypted Blowfish data in the latter case therefore extends the size of the encrypted data to 384 bits in length. The results for this are shown in Figure 6-15.

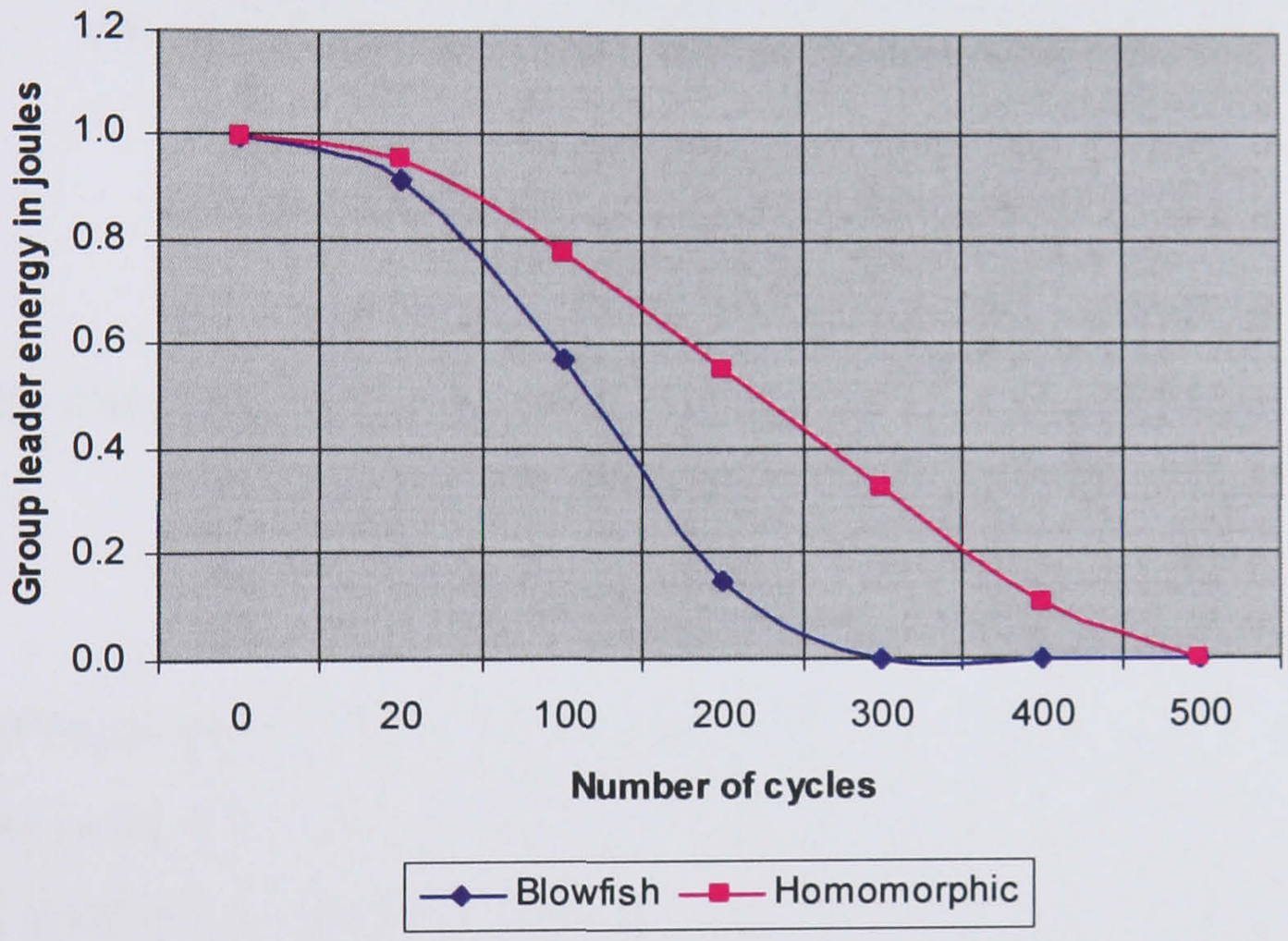


Figure 6-15: Cost of secure data aggregation using Blowfish and Homomorphic encryptions.

Figure 6-16 shows the energy consumptions of the group leader in Cases 1 and 2 for SADI-GKM with increased group sizes. In all cases the result is established for a 100 cycle run. We can see from this how the load on the group leader increases as the group size increases. This is a clear consequence of the increased data that must be aggregated as the group size increases. However, it also highlights how the prevention of replication

attacks in Case 2 affects the level of the energy used by the group leader sensor node, in comparison to that of Case 1.

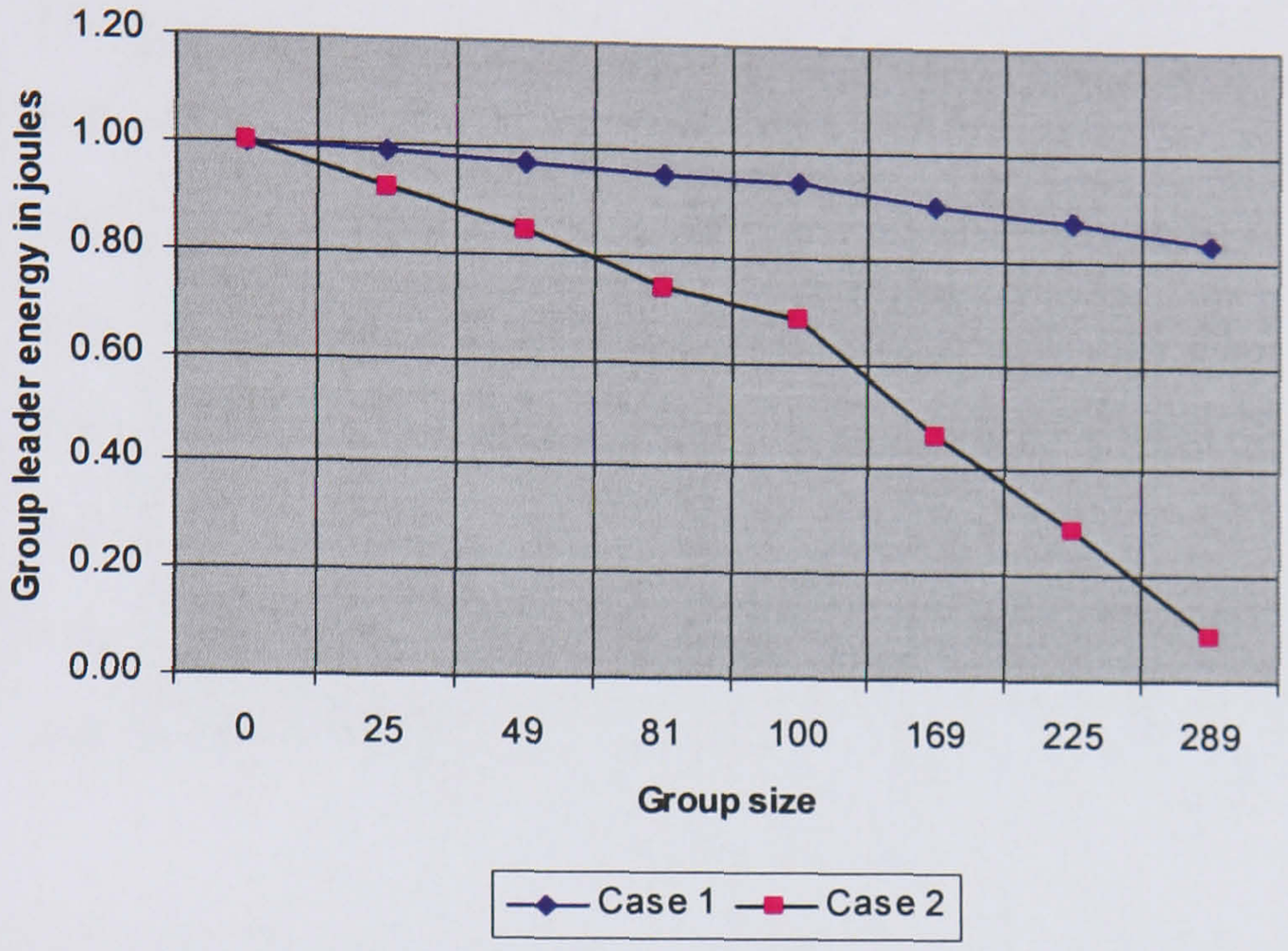


Figure 6-16: Cost of SADI-GKM using different group sizes (100 cycles for each group size).

6.3.4 Costs of Authentication and Data Freshness

In the previous section we have described the implementation of the second layer along with the first layer of our protocol. Since we assumed all sensor nodes are trusted, we therefore didn't make use of authentication and didn't include the cost of authentication in our analysis. In this section we will add the extra cost of authentication and checking of data freshness. In Figure 6-17 we have presented the costs of homomorphic encryption with and without authentication, where the first case is more costly than the second. The reason for the cost increase in the first case is due to the increase in the packet size as a result of adding hash value X_i and time stamp TS_i. The value X_i is produced using the hash of encypted data M_i, value V_{GI} (the authentication value shared by all member sensor nodes), ID_i ,and the time stamp TS_i (used for data freshness and to prevent a replication attack) as shown in Algorithm 6-2. In this experiment, the group size is 100 sensor nodes,

and the packet size 34 bytes (24 bytes of encrypted data, 2 bytes of ID, 4 bytes for the time stamp and 4 bytes for the authentication value X_i).

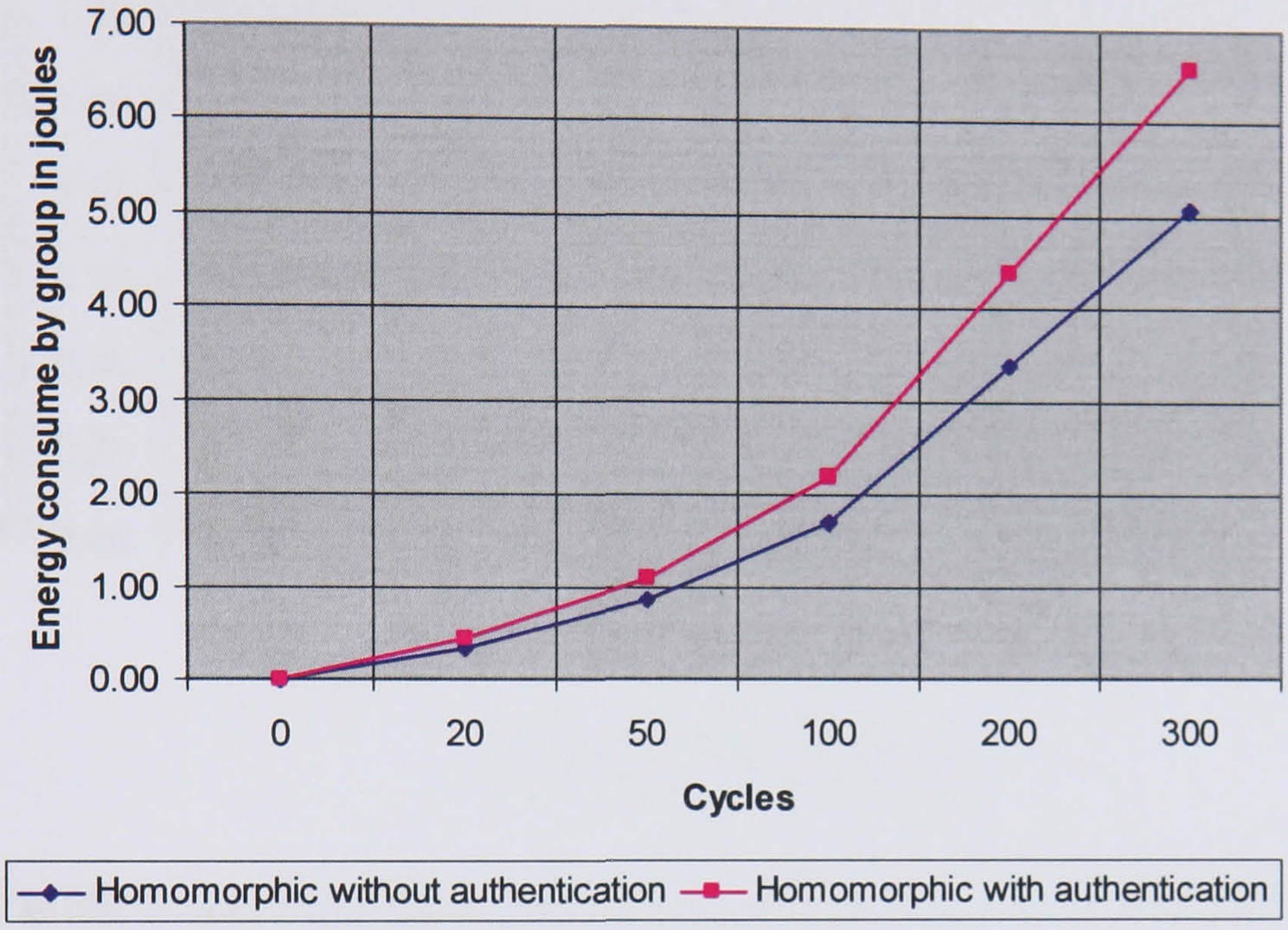


Figure 6-17: Energy consumption of the entire group with and without authentication using Homomorphic encryption.

The net cost of authentication for the entire group of sensor nodes after running n cycles is shown in the Figure 6-18 below.

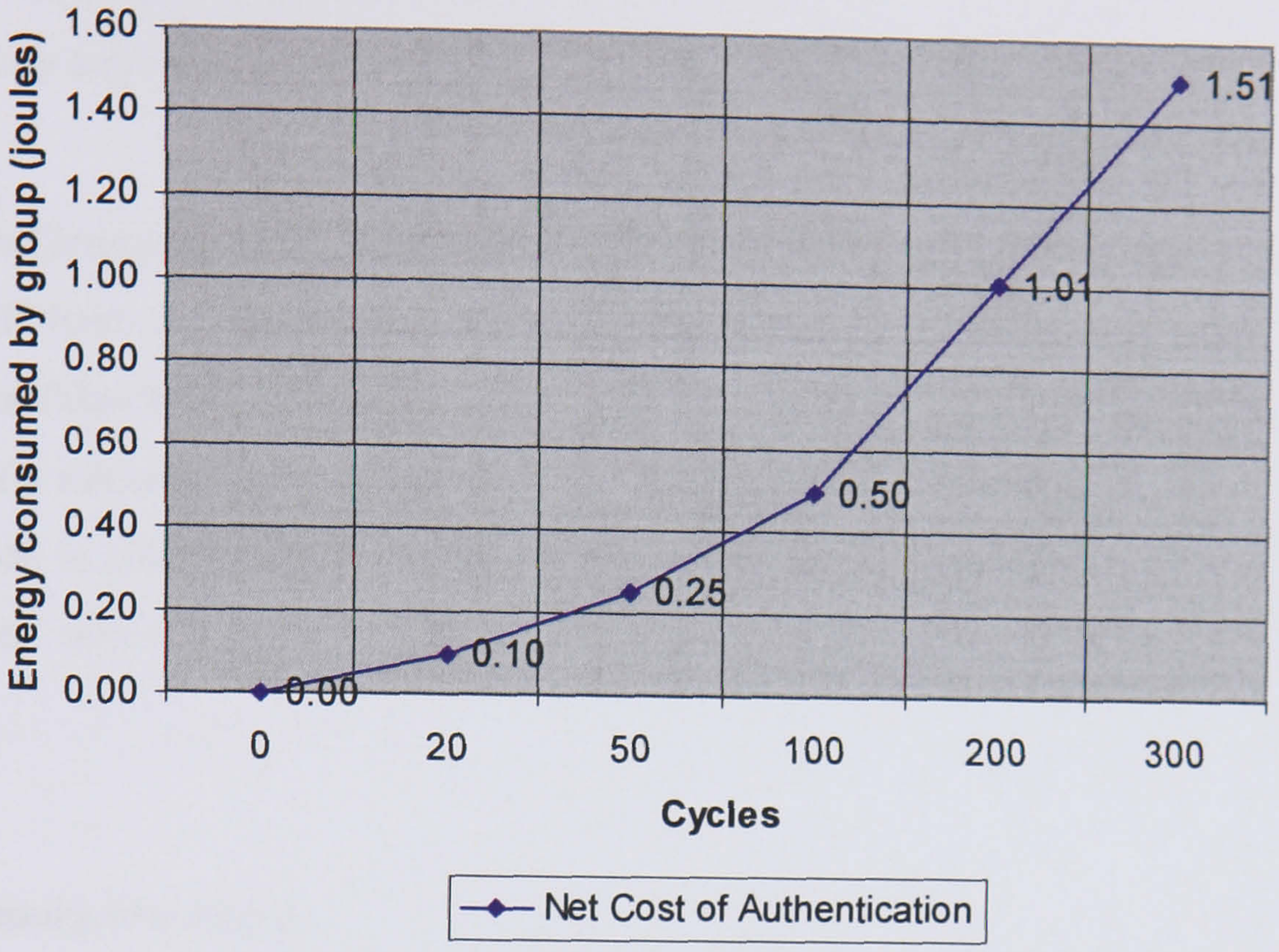


Figure 6-18: Costs of authentication for the entire group of sensor nodes after running n number cycles.

6.3.5 Forward and Backward Secrecy

Our proposed key management protocol is able to provide forward and backward secrecy due to the fact that every sensor node in each group has a different key for data encryption. We can justify this as follows.

Joining a Group: When a sensor node intends to join a group *GI,* it will follow a similar process to that of *"Pre-deployment"* as explained in Section 4.1. The new sensor node will be supplied with a fresh key $K_{GI,IDi}$ for encryption, the value V_{GI} for authentication and a node ID_i. The new node must be deployed within the target group according to the position of the group in the WSN. Once the sensor node has been successfully deployed within the group it will be allowed to communicate with member sensor nodes in that group. Additionally, backward secrecy is achieved since all sensor nodes in the group have distinct encryption keys from the new sensor node's key. Consequently all nodes in the group are independent in terms of data secrecy/confidentiality. The data can only be

encrypted at each source sensor node using its unique key and cannot thereafter be decrypted by any other member sensor nodes except the group leader.

Leaving a Group: Suppose a sensor node dies due to power loss or enemy capture. This action will have no affect on the other sensor nodes in the group in terms of data secrecy/confidentiality. Suppose the enemy has captured the sensor node and has successfully retrieved the keys $K_{GI,IDi}$, V_{GI}, and the sensor node's ID. Although the enemy sensor node is able to launch a replication attack using V_{GI}, it has no access to the data from other sensor nodes during communication, and forward secrecy is therefore achieved.

6.3.6 Memory Overhead

In this section we compare our proposed SADI-GKM protocol with the Group-based EG [74], PIKE [76] and DGKE [163] in terms of memory overhead. The memory overhead for PIKE-2D is $\lceil \sqrt{n} \rceil + 1$ where n is the total number of sensor nodes, and DGKE requires only $n_k \leq d$ keys where n_k is the number of keys and d is the density of sensor nodes. The SADI-GKM protocol is more flexible in terms of memory overhead since every sensor node in the WSN has two keys: one for encryption and another for authentication. The group size doesn't affect memory overhead in our scheme, whereas the number of keys stored in each sensor node by PIKE and Group-based EG increases as the size or number of groups increases. Group-based EG stores 50 keys on average in every sensor node in a group on the basis of a 40m radio range. In DGKE the number of keys depends only on the number of neighbours a sensor node has, but our proposed SADI-GKM protocol is density independent, namely, the number of neighbouring sensor nodes has no effect on memory use.

6.3.7 Connectivity

A particular advantage of our scheme is that we are able to achieve absolute unconditional connectivity in any size of group. Li et al. [80] also achieve 100%

connectivity [135] but their 100% connectivity is conditional. As stated in their paper [80], their scheme achieves full connectivity only when 55 keys are assigned in each group. Additionally, the size of group doesn't affect the performance of our scheme either, unlike those solutions such as the static key pool based idea and other group-based schemes. However the keys used for communication links are different in the key pool based idea as compared to other schemes. We note that the impact of performance of the key pool based idea is reduced for authentication in mobile sensor networks because a mobile sensor node can roam and authenticate itself by checking keys with different host nodes within a group.

6.3.8 Secure Group Leader Selection

We have implemented and undertaken performance evaluations of our group leader selection scheme. We have also performed a comparison of the scheme with other proposed schemes. We have implemented our group leader selection scheme on a grid topology. The packet size used during our implementation is 31 bytes, which include 24 bytes of encrypted data (using homomorphic and Blowfish encryption), 2 bytes for a node ID, 4 byes for a time stamp and 1 byte for an authentication value. We have used the same routing algorithm as described in section 6.2.

As defined in the previous chapter, we have used four different parameters in our group leader selection formula: *energy*, *number of neighbouring sensor nodes*, *distance from the current group leader* and *trust*. Although we include trust in the calculation, we are not proposing any scheme in this project to calculate a trust value for every sensor node. Therefore during the evaluation we have used constant trust values for every sensor node and assumed all nodes are trusted. Any proposed solution for finding the trust values of sensor nodes can be easily integrated into the formula.

For the group leader selection formula we evaluated various cases, including:

Case 1: In this case we have considered a group of 25 sensor nodes. The topology we have used is a grid topology with a fixed distance of 20 meters between sensor nodes. The initial energy of all sensor nodes is set to one Joule. Consequently the total initial

energy of a group is 25 Joules. As described in Chapter five, in our group leader selection scheme we only involve the neighbouring sensor nodes of the current (outgoing) group leader in the selection process. In contrast, existing proposed schemes consider the entire group in the selection process. To make an appropriate comparison, we have also considered a similar process within our SADI-GKM third layer, allowing us to test the case of involving all sensor nodes in the selection process as well as the case of just involving neighbouring sensor nodes. During this experiment the group leader selection process will take place ten times, each of which happens after a fixed number of cycles. One cycle constitutes all sensor nodes in a group sending one packet to the group leader. When one cycle completes the group leader will have received 24 packets from all member sensor nodes plus one packet from the group leader sensor node itself and send it toward the sink. Among other group leader selection/election schemes, LEACH [139] only considers energy as a factor during group leader or cluster head selection, whereas Wen et al. [140] considers both energy and the number of neighbouring sensor nodes in the selection process. We consider LEACH and Wen et al.'s since both are well established and widely known protocols for group leader selection in WSNs. In Figure 6-19 we have presented results for LEACH, Wen et al.'s scheme, the *SADI-GKM All* scheme (meaning that all sensor nodes participated in the selection process) and the standard SADI-GKM scheme (for which only neighbouring sensor nodes participated in the selection process).

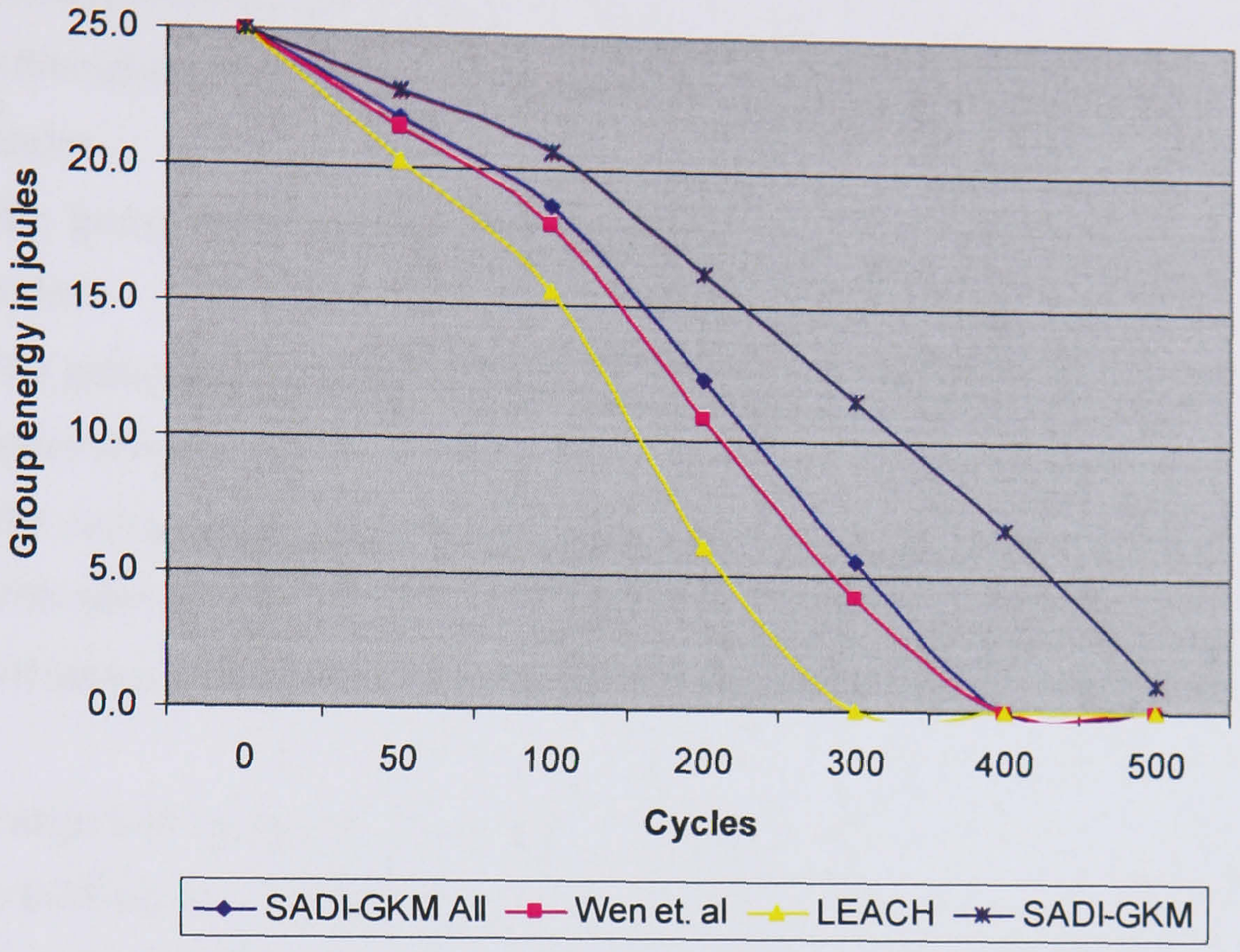


Figure 6-19: Group energy consumptions with 25 sensor nodes for n cycles.

In the first set of experiments we have evaluated all the four methods for group leader selection using 50 cycles. After every 50 cycles a new group leader is selected and this process will be done ten times. In other words, once the group leader receives 1250 packets (25×50) it will initiate the new group leader selection process. Furthermore, ten times group leader selection process will run where each selection starts after fifty cycles. In the second set of experiments we have considered 100 cycles. In this case the new group leader selection process will start after 100 cycles (100×25 = 2500 packets received by the current group leader). Furthermore the group leader selection process will run ten times, where each selection is initiated after 100 cycles of the simulation. During all these experiments we have considered the energy cost of group leader selection. There are certain numbers of communication steps that happen during the group leader selection process, as described below:

- The group Leader sends a *selection process* packet to all/neighbouring sensor nodes.

- All/neighbouring sensor nodes receive the *selection process* messages.
- All/neighbouring sensor nodes send their weighting factors to the current group leader.
- The group leader receives all weighting factors from all/neighbouring sensor nodes.
- The group leader sends the selected new group leader ID to all member sensor nodes sensor nodes.
- The current group leader will send information (the master key and keys shared with other groups) to the new group leader.
- All sensor nodes receive the new group leader ID.

All these steps will remain the same for the rest of the cases.

In Figure 6-20 we can see that SADI-GKM has consumed less energy as compared to all the other schemes to prolong group life time. Results in Figure 6-20 demonstrate that using LEACH and Wen et al, entire group energy runs out after 10 cycles (10×100 = 10,000 packets) and using SADI-GKM entire group energy finish after 25 cycles (25×100 = 25,000 packet). It means SADI-GKM shows 60% better performance to prolong group and WSN life as compare to existing schemes. One of the main reasons for SADI-GKM reduced energy consumption is due to the fact that fewer sensor nodes need to participate in the selection process. As pointed out in Chapter five, packet and group sizes have a direct impact on the energy consumption for the group leader selection process. This can be further supported by the next case where we have increased the group size.

Case 2: In this case we have considered a group of 100 sensor nodes. As with Case 1, we have used a grid topology with a fixed distance of 20 meters between nodes. The initial energy of each sensor node is one Joule. Consequently the total initial energy of the group is 100 Joules. During the experiments of Case 2 the group leader selection process will take place 50 times after a fixed number of cycles. In this case one cycle means all sensor nodes in the group will send one packet to the group leader sensor node. When

one cycle completes the group leader will have received 99 packets from its member sensor nodes plus one packet from the group leader itself for data aggregation.

In the first set of experiments we have evaluated the four methods for group leader selection using 5 cycles. This means that after every 5 cycles a new group leader sensor node will be selected. Once the group leader sensor node has received 500 packets (100×5) it will initiate the new group leader selection process. In the second set of experiments we have considered 10 cycles, so that the new group leader selection process starts after every 10 cycles (10×100 = 1000 packets received by the current group leader). This experiment was run until 50 rounds of new group leader selection had occurred.

In Figure 6-20 we can clearly see the effect of the larger group on the group leader selection. Using SADI-GKM All, LEACH and Wen et al.'s method, the entire group energy was consumed after just 15 cycles. The reason for this higher energy consumption is that 50 rounds of new group leader selection are applied, in addition to the larger group size. Most of the energy was consumed during the selection processes. Therefore we must be particularly careful about how to operate the selection process for large groups. From Figure 6-20, it is clear that our SADI-GKM is more energy-efficient.

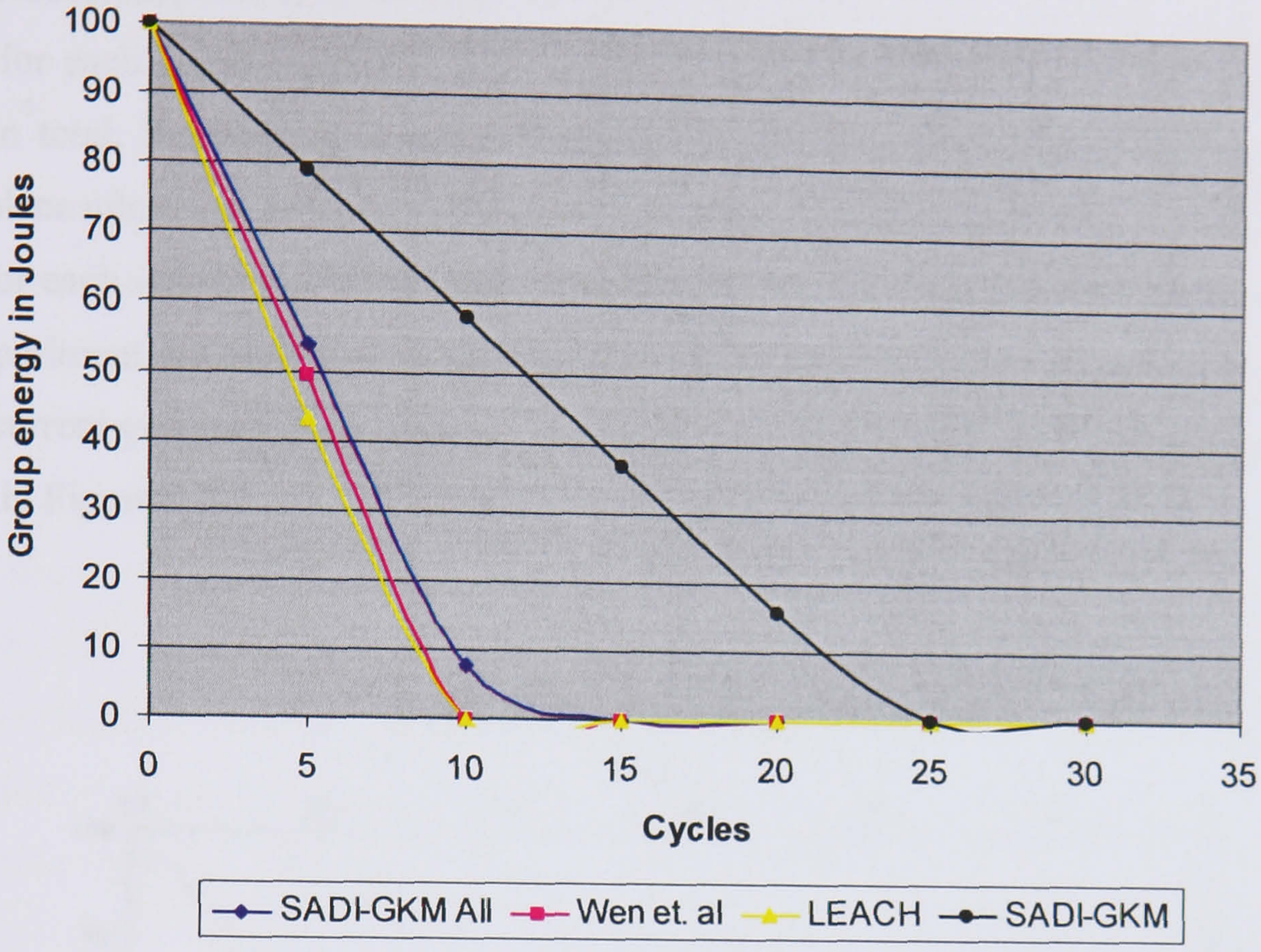


Figure 6-20: Group energy consumption with 100 sensor nodes using different schemes.

Another finding from the experiments is that we need to find a careful balance between the numbers of group leader selection rounds and cycles. It would be unwise to start a new group leader selection process after just a few cycles, as the group leader selection process costs significantly in terms of energy. Therefore we should find optimum values, which will be different for different group sizes. In Case 3 below we have run 60 different experiments in an attempt to establish different energy consumption figures.

Case 3: In this case, the group size is 100 sensor nodes, the total initial energy of the group is 100 Joules, and the packet size is 31 bytes. For each experiment we run m cycles for each selection process, with the selection process occurring n times in total, written as (m, n) where $m=61-n$. In this experiment we have started $m = 60$ and $n = 1$. Furthermore we reduce m by one and increase n by one in each experiment. For example in the first experiment (60, 1) we have run 60 cycles for each selection process, with the selection

process occurring one time in total. In the second experiment (59, 2) we have run 59 cycles for each group leader selection process, with the selection process occurring two times in total. Similarly in the third experiment (58, 3) three selection processes have taken place after each 58 cycles. This continues until (1, 60), which means we have run 1 cycle for each selection process, with the selection process occurring 60 times in total. In this experiment we have only used SADI-GKM, i.e., only the neighbouring sensor nodes of the current group leader participate in the selection process. The experiment results are shown in Figure 6-21.

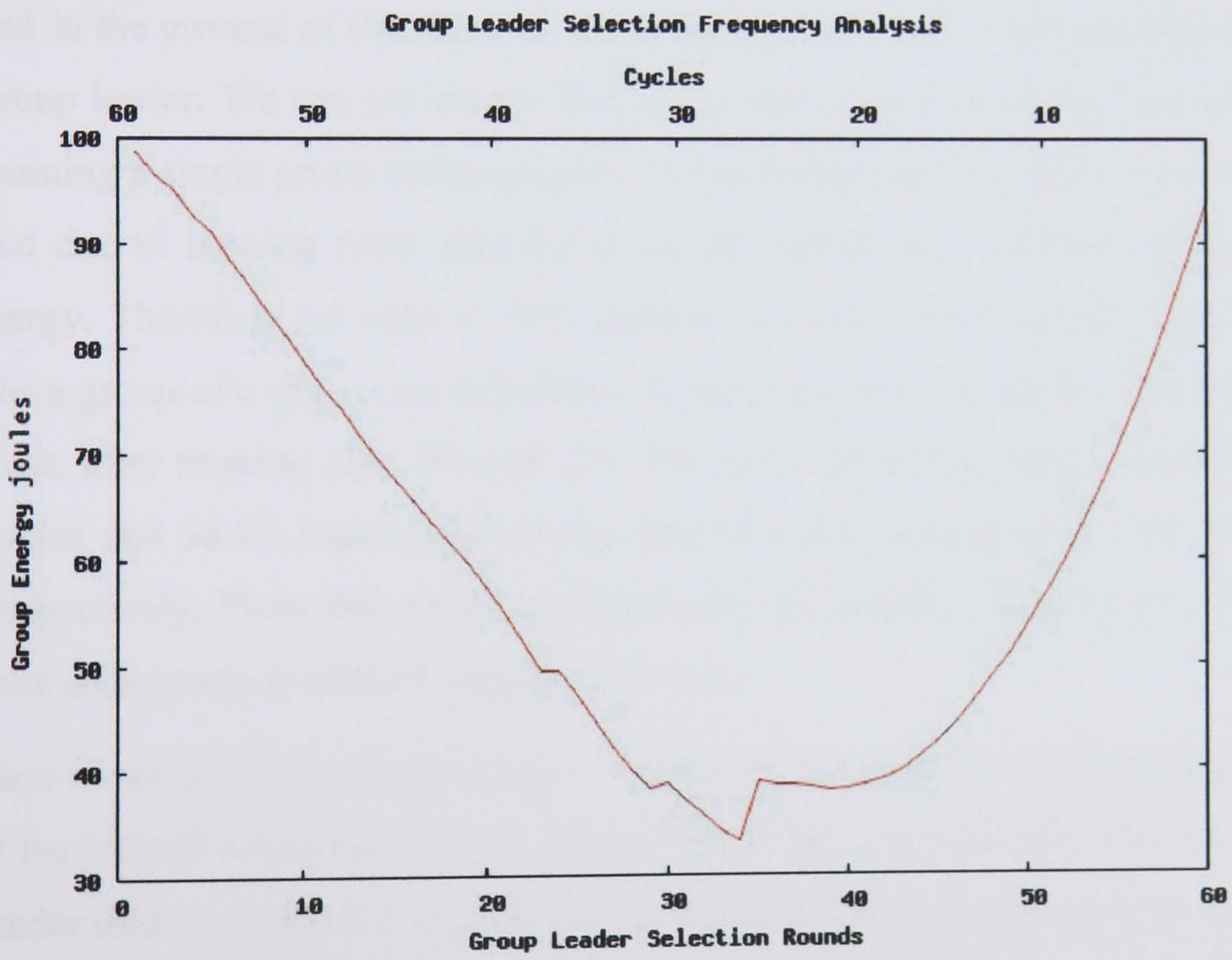


Figure 6-21: Energy consumption for numbers of group leader selections versus numbers of cycles.

Using this experiment we can establish the optimum values for the number of group leader selections versus the number of cycles to reduce energy overhead for any group size. To understand the results in Figure 6-21, we use both Figure 6-21 and Figure 6-22 for explanation. From these we can identify the optimum values of group leader selection

and cycles for a group of 100 sensor nodes. The main aim is to select optimum values which can provide a longer overall group lifespan, or maximum output from the group in terms of the amount of data sent. In Figure 6-22 we can see the number of data packets sent to the group leader. However these cumulative packets do not include the packets sent out during the selection process. In Figure 6-21, for the result for (1, 60) where after each cycle (1 cycle is equivalent to 100 packets sent) a group leader selection process runs and this process will run 60 times, the final group energy was 93.8 Joules (the total energy consumed in the process of (1, 60) is 6.13 Joules) and a total of 6094 packets were sent to the group leader. However for (60, 1) where the group leader selection process runs one time after sixty cycles, the final group energy was 98.7 (the total energy consumed in the process of (60, 1) is 1.23 Joules) and a total of 11404 packets were sent to the group leader. We can see clearly that in the latter case less energy was consumed due to running a single group leader process. In the former case (1, 60) more energy was consumed due to running extra selection processes where each selection process costs extra energy. Therefore we need to find optimum values of group leader selection and cycles for a group of a given size to balance energy consumption and prolong group life. In contrast, after running (31, 30) and (26, 35), the final group energy remaining was 38.75 Joules and 38.92 Joules respectively, and the total packets sent were 95704 and 94154 respectively. From this we can establish that the highest performance value for a 100 sensor node group is with 31 rounds/30 cycles.

The reason we want to balance the cycles with the number of group leader selections is to increase the overall availability of the group. This is because running many cycles on a group leader node can increase the chances of it dieing and encouraging such action can cause more sensor nodes to die, which will lead the entire group to the point where sensor nodes will be unable to communicate with the rest of the sensor network even though there will still be some active sensor nodes. Therefore we need to create a balanced energy consumption across all of the sensor nodes to increase the overall group life. This was a major motivation behind our proposed solution. In other words even though selecting a new group leader costs energy it is still better as compare to single sensor node to be a group leader for longer period which misbalances the energy consumption of

entire group and results in more energy lost. This argument is discussed with help of Figure 6-22.

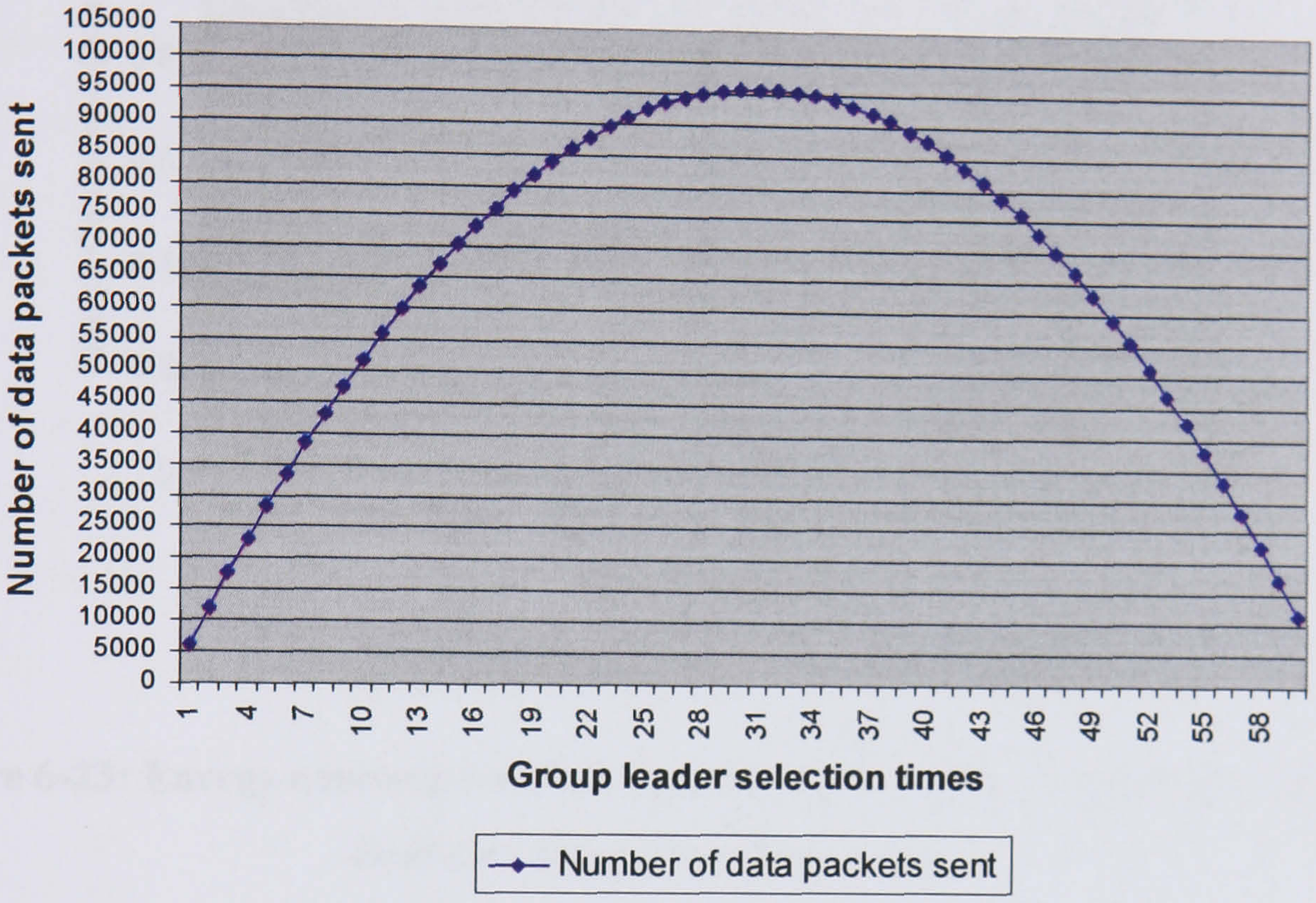


Figure 6-22: Number of data packets sent after n number of group leader selection process.

The experiments presented so far are for the energy consumptions of one large group. We have also conducted another set of experiments for a comparison of energy consumptions between a single group and multiple sub-groups. This shows a significant difference as indicated in Figure 6-23. During the first experiment we have used one group of 36 nodes and in the second experiment, 4 subgroups with 9 sensor nodes for each. In the first experiment data aggregation takes place at a single group leader sensor node, whereas in the second experiment all the four subgroups aggregate data locally and send them to the main group leader sensor node which further aggregates the data from the four subgroup leaders. The main reason for the reduction of energy consumptions in the second experiment is due to the reduction in the number of transmissions towards the main group leader sensor node. This set of experiments has been run using different cycles (every single sensor node senses an event and sends it towards the group/subgroup leader).

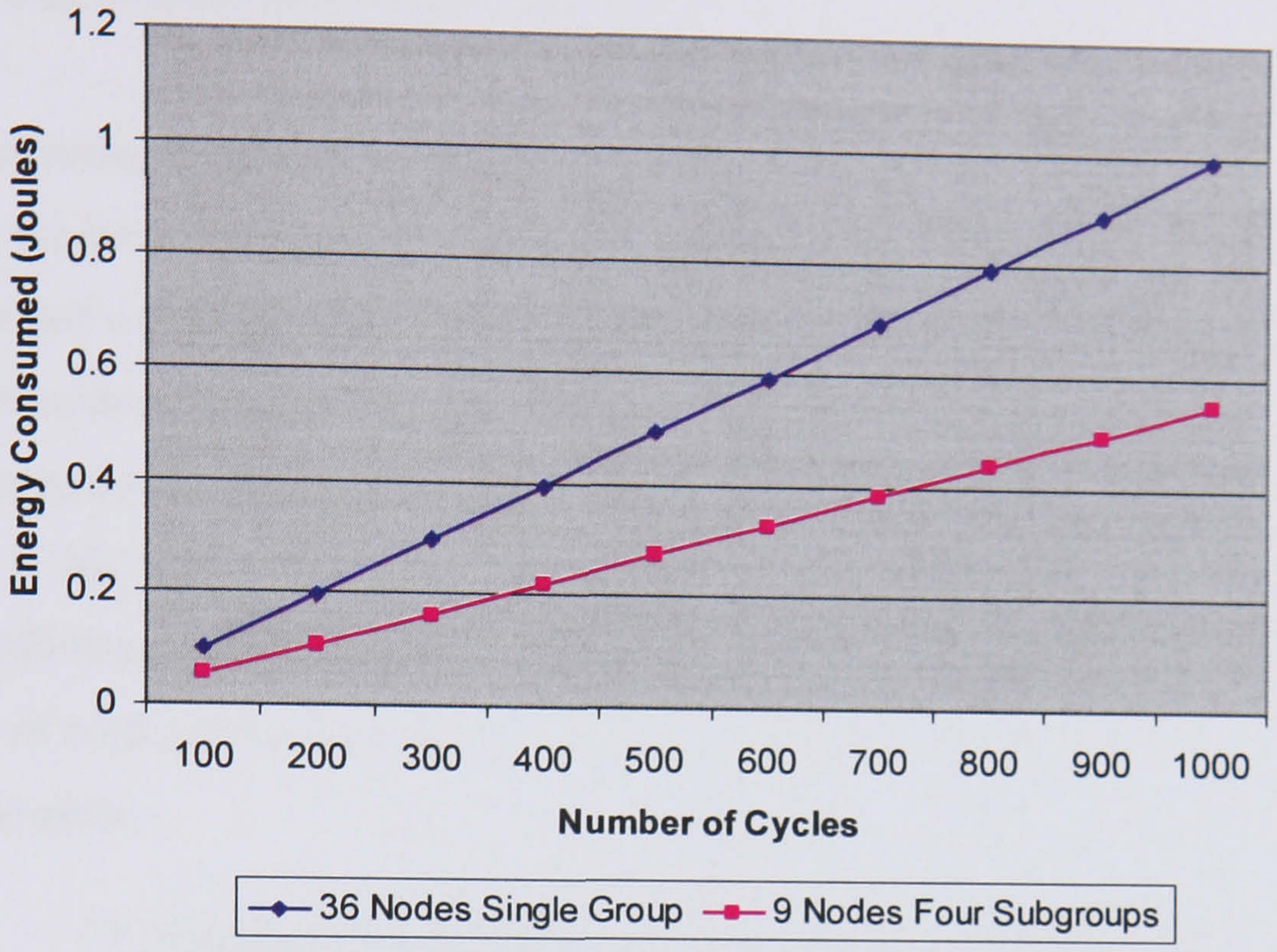


Figure 6-23: Energy consumptions for data aggregation using a single group and multiple subgroups respectively.

6.3.9 Key Management for MSNs

In Chapter five we proposed a key management scheme for MSNs based on SADI-GKM. According to SADI-GKM, nodes in a group are unable to communicate directly with those in other groups, in order to contain the spread of potential damage caused by attacks. This raises the question of how a mobile node in MSNs can authenticate itself when it roams from one group to another. Therefore Layer 4 incorporates additional functions which include key configuration to allow mobile sensor nodes to authenticate themselves.

In this section we need to find the probability of key sharing in the host group when a guest mobile sensor node roams into it. It is likely that the mobile node will have more than one sensor node within its range in the host group. Whether the mobile node is allowed to communicate with only one or more neighbouring sensor nodes depends on the security policy in place. In the previous chapter we presented a formula to find the probability that a mobile sensor node will share a secure communication link with at least one sensor node, given a sensor node density of *d*. To validate the formula, we have

carried out experiments for the following cases:

Case 1: Increasing the Number of Nodes in a Host Group

In this case we have found the probability of establishing a secure link between a mobile sensor node and a member sensor node in its host group. As described in section 5.2.3.2, every sensor node is assigned 75 keys from the key pool for authentication and separate keys for encryption. In this experiment we have assumed that the geographical area of the host group is 200×200 m^2 and the radio range of each node is 20 meters. Figure 6-24 shows the different probabilities of establishing links for different group sizes. Increasing the number of nodes in the host group will increase the probability of key sharing for the guest mobile node.

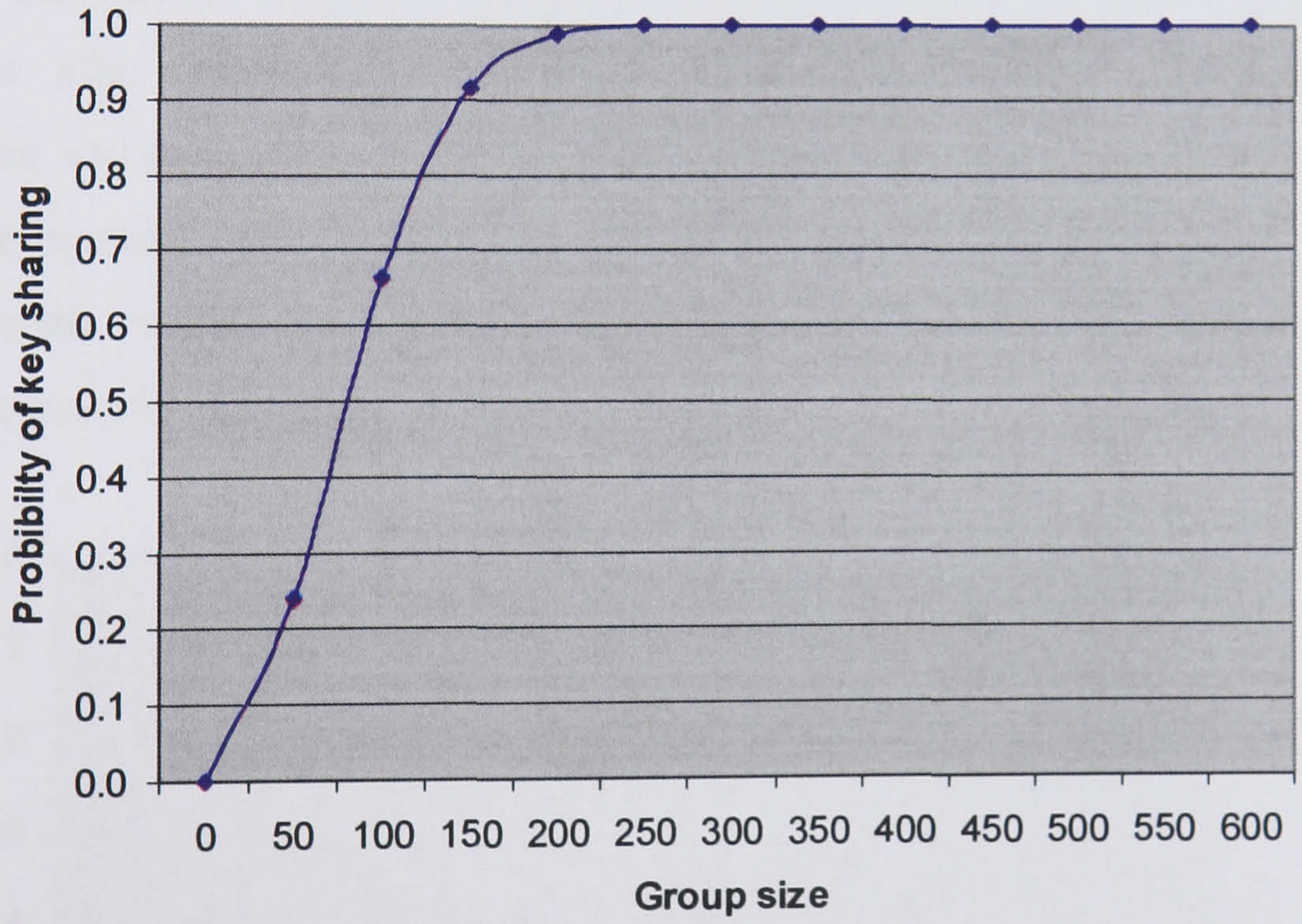


Figure 6-24: Probabilities of establishing links for a mobile sensor node in different host group sizes.

Case 2: Probability of a Mobile Node in a Host Group during Roaming

In this case we need to find the probability for a mobile sensor node to roam from one side to another side of its host group. In this experiment the radio range of each sensor node is 20 meters, the geographical area of the host group is 200×200 m^2, there is a total

of 100 nodes in the host group, and the speed of the mobile node is 1 meter/second. As the mobile node moves to the centre of the group the density (number of neighbouring nodes) of the mobile sensor node increases, which also increases the probability of key establishment being possible. During this simulation we have deployed 100 sensor nodes randomly into the 200×200 m^2 group area, and then checked the probability that the mobile sensor node at its current position is able to share at least one key out of density *D* (neighbouring sensor nodes). For every position of the mobile sensor node (moving in from one corner of the group and moving out from the opposite corner) we have run the simulation 10,000,000 times, each time we have also changed all sensor node positions in the group to find an accurate probability. Figure 6-25 shows the results from these experiments, where we can see that if the mobile sensor node is close to a boundary edge (e.g. when the mobile sensor node roams into the group) it has the probability 0.3. Furthermore after roaming 35 meters inside the group the probability of key establishment increases to 0.75. Similarly when the mobile sensor node roams towards the opposite boundary of the group the probability of key establishment again reduces. Furthermore the energy cost of the host group will increase when more guest mobile nodes arrive into the host group.

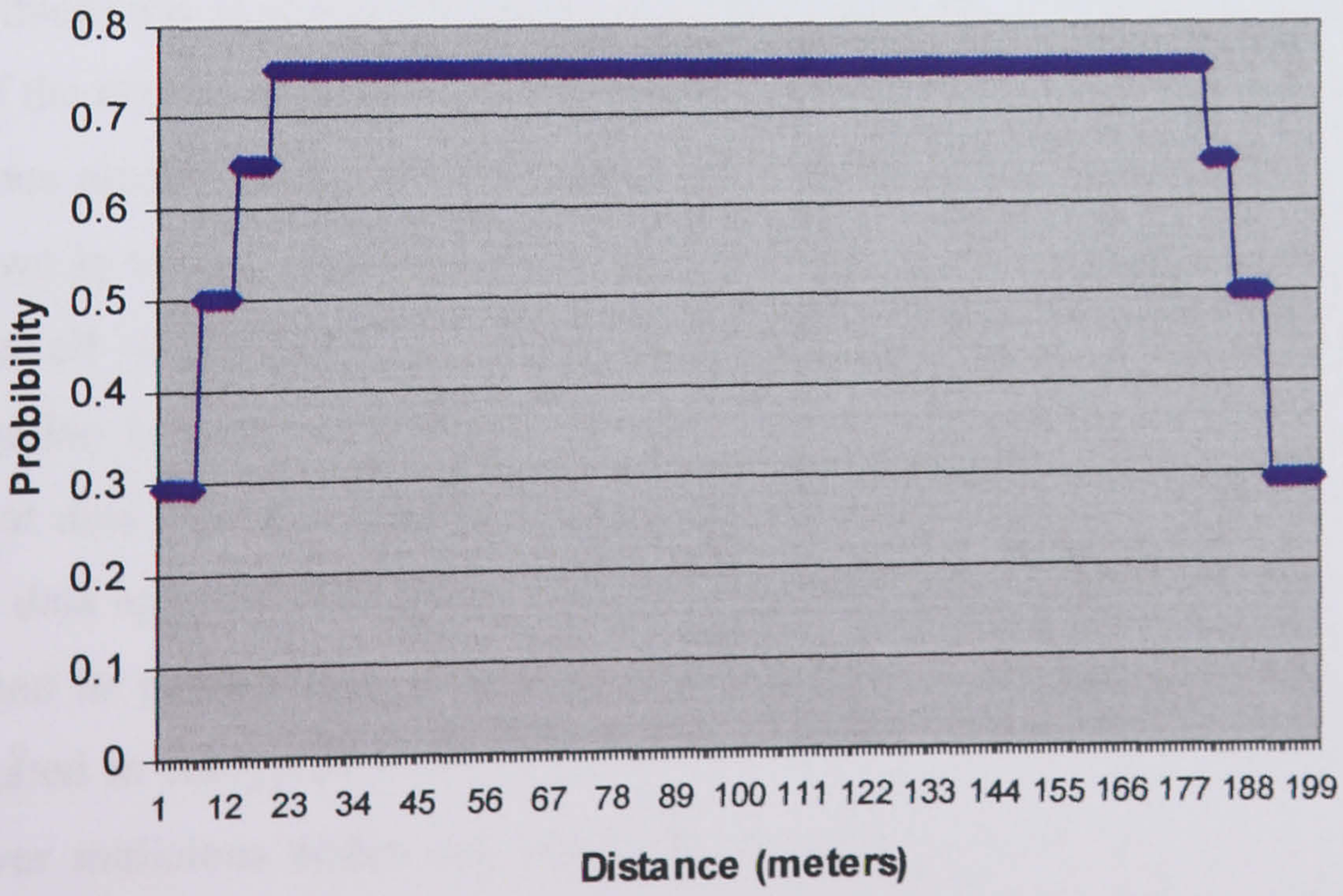


Figure 6-25: Mobile sensor node probabilities of key sharing with host group member nodes during roaming.

6.4 Discussion

As we know WSNs is scalable and a type of ad hoc networks where nodes will frequently join or leave due to energy depletion or mobile nature. This brings frequent changes in topology which affect the security of entire WSN. Our simulations suggest that SADI-GKM is topology and density independent and changes in topology will therefore has less effect as compare to other schemes. In particular, the structure and density independence helps to reduce computation costs in maintaining the network topology when new nodes join or leave a group/network for example in mobile sensor networks.

In considering security we focus especially on node capture attack due to its unique nature and its strong post-attack effects on the entire WSN. In such an attack the adversary may be interested to discover more information from the WSN by physically compromising a sensor node, reprogramming the sensor node and placing it back in the WSN as a genuine sensor node. This presents a high risk for data confidentiality and integrity. Due the nature of node capture attack it is very hard to provide a perfect solution. However its consequences can be minimized. We have studied and analysed node capture attack behaviour in different scenarios which makes us able to establish some precautions, as mentioned in section 5.1, to provide better resilience. We believe in such attacks the user will always prefer to protect their data's confidentiality at any cost even if the node is physically compromised. SADI-GKM has successfully achieved better resilience against node capture attacks by keeping data confidentiality at a high priority as shown in section 6.3.1. However it is impossible in a scalable application to keep and process all of the data in its encrypted form to achieve high confidentiality. Data aggregation is required in order to reduce the communication overhead. To reduce the risk that data can be comprised during the aggregation process SADI-GKM provides a secure data aggregation scheme using homomorphic encryption as mentioned in section 5.2.1 and its performance is shown in section 6.3.3. In homomorphic encryption data is aggregated in encrypted form which helps to maintain data confidentially and integrity. However malicious nodes can send false encrypted values to disarray the aggregated results. It remains an open challenge as to how to authenticate or check data in its encrypted form.

Similarly if an aggregator node is comprised or stops functioning due to depleted energy, there should be a mechanism to select or elect a new aggregator node to keep the group connected with the rest of the WSN. The selection process needs to be energy efficient and secure. In case an adversary node is selected as an aggregator or group leader it will not only compromise the communication of this entire group but also try to disorder the rest of the network. SADI-GKM provides an efficient group leader selection scheme as compared to existing schemes as shown in section 6.3.8. However in MSNs if a group leader is mobile it will increase communication overheads and also make the selection process difficult. Consequently security challenges in MSNs are more difficult to handle. Therefore we have extended SADI-GKM toward MSNs. However security in MSNs has many important unexplored challenges and many open research issues.

6.5 Summary

In this chapter, we have presented the implementation phases, simulation framework, analysis, results and performance evaluation of our protocol. First we have explained the implementation phases including topology implementations, routing algorithms and security. Subsequently we have evaluated the SADI-GKM performance against node capture attacks using various topologies, both with and without the use of groups. We have also described our implementation of secure data aggregation using homomorphic encryption, secure group leader selection and finally key management for MSNs. We have demonstrated through simulations that our proposed protocol has good resilience against node capture and replication attacks compared to other existing schemes, and that our novel group leader selection scheme shows better performance than existing group leader selection schemes.

Chapter Seven: Conclusion and Future Work

This thesis has presented a new key management protocol, SADI-GKM, to fulfil the proactive security needs of WSNs. This protocol is an integration of different novel layers (basic key management, secure data aggregation, secure group leader selection and key management) that we have developed during our research. The aim of the protocol is to provide structure and density independent key management for large scale WSNs, which can provide secure communication between source and destination nodes, resilience against node capture and replication attacks, secure data aggregation, secure group leader selection and key management for MSNs.

This chapter provides a summary and conclusion of our work together with future research in the subject area. It is organised as follows. First we present a summary of the thesis in Section 7.1. A summary of the SADI-GKM protocol and our main contributions are presented in Section 7.2. The comparison of SADI-GKM with existing approaches is discussed in Section 7.3. Future work is investigated and proposed in Section 7.4, and finally our concluding remarks are provided in Section 7.5.

7.1 Thesis Summary

WSN development is an exciting research area due to the constraints involved. The reason for the popularity of WSNs is due in part to the small sizes and low costs of sensors, their operations and networking behaviours, which enable them to provide significant advantages for many applications that would not have been possible in the past. Battlefield surveillance, forest fire detection, smart environments and environmental control in office buildings are well known examples of their applications.

A WSN is composed of a large number of sensor nodes that are densely deployed either inside the phenomenon or very close to it. The positions of sensor nodes need not be

engineered or predetermined. This allows random deployments in inaccessible terrains or during disaster relief operations. On the other hand this also means that WSN protocols and algorithms must possess self-organizing capabilities. Another unique feature of WSNs involves the cooperative efforts of sensor nodes. Sensor nodes are fitted with on board processors and they can use their processing abilities to locally carry out simple computations and transmit only required and partially processed data. In the future WSNs will form an integral part of our environment and lives.

There are two main components in a WSN: *sensor nodes* and the *sink*. Sensor nodes can be categorized as electronic sensors, electronic-portable or low cost sensors and mechanical sensors. All these types of sensor nodes have different sensing capabilities.

The sink in a WSN can be a computer, laptop or a sensor node which gathers information or data from the other sensor nodes and provides this to users or forwards it to other networks, such as a local ad hoc network or the Internet. In other words the functionality of the sink in a WSN is similar to that of a server in a traditional network. In almost all WSNs data are routed toward the sink and the hops close to that sink become heavily involved in packet forwarding and thus their batteries get depleted rather quickly.

WSNs are different from traditional networks and present a new set of properties. Typically the structure of a traditional network will remain the same in all its applications while a WSN's structure will change according to its application.

Secure communication between network components is always an issue. Researchers are continuously inventing new methods to provide more and more secure forms of communication. Many key management protocols have been developed in the past for traditional wired and wireless networks. Transfer of networking technology from wired to wireless increases potential security threats, ranging from passive eavesdropping to active impersonation, message replay and distortion. In WSNs we have the same challenges, but in addition the limited resources of sensor nodes pose the biggest challenge. As a result, we are unable to directly use traditional networking techniques for WSNs. For example, asymmetric cryptography may need to be avoided wherever possible due to its demands on processor resources.

Our work focuses on the design of a structure and density independent key management protocol stack (SADI-GKM) for large scale WSNs. This protocol can be used as a proactive security solution for WSNs, which consists of different layers for different security services. All these layers are integrated with each other to provide better secure communication between source and destination nodes. SADI-GKM provides better resilience against node capture and replication attacks, secure data aggregation, secure leader selection and key management for MSNs using fewer resources.

In this thesis we have presented our work on developing and evaluating the SADI-GKM protocol, and justifying the above capabilities. In order to achieve this we included the following materials:

Our introduction to the area in Chapter 1 discussed the wider context and outlined the problem of secure communication in large scale WSNs. It includes the definition of WSNs, their main applications and current WSN projects. It also describes the communication architecture and components of sensor nodes. We also briefly describe the importance of security in WSNs. Chapter 1 also highlights the consequences of node capture attacks, other interlink attacks and security vulnerabilities.

In Chapter 2 we have given a general overview of challenges in WSNs, including: fault tolerance, sensor network topology, routing, mobility and scalability. Furthermore we have specifically describes the challenges and possible attacks in WSNs which include: data confidentially, data integrity, authentication, key establishment, availability, privacy, secure routing, secure group management, intrusion detection and secure data aggregation. At the end of chapter we have given an overview of security in MSNs.

In Chapter 3, we have surveyed the literature and related works relating to key management, secure data aggregation, group leader election/selection and key management for MSNs. We have presented and discussed the existing solutions of key management for static WSNs. These solutions were classified into five different types includes: key pool based key management, session based key management, hierarchical based key management, key management for heterogeneous sensor networks and group based key management. All these solutions place emphasis on the important issue of providing high resilience against node capture attacks and providing better secure

communication between source and destination. This chapter has pointed out the main drawbacks of existing key management solutions. We found that all these solutions are structure dependent and any change in the structure directly affects the security of the WSN. Furthermore in key pool based key management a few compromised sensor nodes could lead to the compromise of the entire WSN. Hierarchical based key management solutions are more prone to node capture attacks, and session based key management solutions are not scalable.

Later in the chapter we have described related works concerning secure data aggregation and its importance. Secure data aggregation is vital in applications where very sensitive data are communicated through sensor nodes. The compromise of an aggregator node can be a significant risk to data confidentiality. Therefore data aggregation should be performed on encrypted data without decryption to improve data confidentiality. Furthermore we have presented related works about secure group leader selection and also the work on key management for MSNs.

Chapter 4 presents the design of our protocol SADI-GKM including its four layers. First it describes the background of the protocol design to highlight necessary requirements. Subsequently it presents our research objectives that form a comprehensive set of schemes. We have also identified issues and challenges that are important when designing an effective key management protocol for large scale WSNs. This chapter highlights the importance of providing an integrated proactive security solution and explains how our four protocol layers are integrated with each other in order to achieve this.

In Chapter 5 the different layers of our protocol stack have been fully described in detail. We began by describing our pre-design investigations. This section provides a detailed analysis for identifying some core information needed to help us toward the development of an efficient and improved protocol design. In the detailed design section we described each layer of our protocol. We began by looking at the first and second layers, *key management* and *secure data aggregation,* where we described the basic key management and secure data algorithms together. Then we explained the third layer,

secure group leader selection, where we have presented our novel formula for new group leader selection. In the fourth layer we presented a key management solution for MSNs.

Chapter 6 presents the implementation phases, simulation framework, analysis, results and performance evaluation of our protocol. First we explained the implementation phases including topology implementations, routing algorithms and security. Subsequently we evaluated the SADI-GKM performance against node capture attacks using various topologies, both with and without the use of groups. We then described our implementation of secure data aggregation using homomorphic encryption, secure group leader selection and key management for MSNs.

Finally, suggestions for future work and conclusions are presented in this chapter.

7.2 Comparison with Existing Approaches

As mentioned in Chapter 4, the main objective of our proposed protocol SADI-GKM is to provide structure independent proactive security solutions for WSNs. The four layers of SADI-GKM offer high resilience against node capture and replay attacks, better data confidentiality, more efficient secure group leader selection and key management for MSNs. SADI-GKM is a component based protocol and easily extendable through the addition of more layers. Its current four layers are integrated with each other as a chain, making it more effective overall. In contrast, current solutions are more focused on single problems.

The most notable key management solutions for WSNs are probabilistic key management solutions [72][75][76][78][125][126][128] and group key management solutions [77][130][145]. All of them have taken node capture attacks as a high priority. In these solutions the physical compromise of a few sensor nodes can help an adversary to compromise the communication of the entire WSN. In contrast, SADI-GKM provides better resilience by avoiding key sharing (used for encrypting and decrypting data) with neighbouring sensor nodes. We have compared the resilience of SADI-GKM against node capture attacks with probabilistic and group key management results. SADI-GKM

has shown better performance against these existing schemes, while using fewer resources.

In addition we have improved the performance of SADI-GKM by applying secure data aggregation. We have implemented secure data aggregation using Blowfish and homomorphic encryption. Furthermore we have implemented the third layer of our protocol for secure group leader selection. For this we have used four selection parameters: *energy*, *number of neighbouring sensor nodes*, *position of a node in a group* and *trust*, whereas existing schemes only consider energy and/or the number of neighbouring sensor nodes. Our comparison through simulation shows that our SADI-GKM group leader selection scheme is more balanced and uses fewer resources as compared to other schemes. Finally we have presented simulation results for key management in MSNs.

7.3 Thesis Contributions

In this section we have organised our contributions in two parts. First we present those contributions which were achieved during the pre-design analysis of current WSN security issues. In the second part we describe our remaining contributions. This thesis contributes primarily to the field of security and key management in WSNs.

Contributions – A

This part includes two main novel contributions:

- The first is the finding that the topology of a WSN, the density of sensor nodes and the level of key sharing among neighbouring sensor nodes used for encryption and decryption have a direct relationship with the security of the WSN. For example, a tree topology is less secure as compared to a grid or random mesh topology when it is subject to node capture attacks. Additionally, current key management schemes are structure dependent and any change in the structure of the WSN directly impacts on the security. Furthermore sharing keys for

encryption and decryption with neighbouring sensor nodes can put data confidentiality at risk, especially if there is a high threat of node capture attacks in a specific application where an adversary can compromise sensor nodes physically. Careful selection of these three parameters before proposing any security scheme can reduce the risk of large-scale damage that might result from node capture attacks [33]. The results may be of significant benefit to the future development of WSN security technologies.

- The second contribution is the finding that the position of a group leader has a direct effect on the performance of the sensor group and can increase the communication overhead dramatically. Therefore it is important that we should be careful and consider group leader positions during the selection of a new group leader. We have presented related experimental results in Chapter five.

Contributions - B

We have proposed a novel key management protocol stack consisting of different layers which are integrated with each other. This protocol provides proactive security solutions to different threats in WSNs. Specifically our novel contributions include:

- The first layer of our protocol represents an important contribution to the subject are. This layer has responsibility to pre-establish keys between sensor nodes and provides basic rules and regulations which are further integrated with all the other layers. This key management layer further operates in two phases: a key pre-establishment phase and a data transmission phase. These two phases are described in detail in Chapters four and five. Furthermore this layer is topology independent, i.e., it can work with various topologies. This has been evaluated, without integration with the other layers, using different topologies both with and without group structures, and compared against existing key management schemes. The comparison results show a significant improvement in resilience against node capture attacks, memory overhead and connectivity.

- As described in earlier chapters, WSNs are highly scalable networks, deployed with potentially thousands or even greater numbers of sensor nodes. Consequently, a potentially huge amount of communication may happen during any operational task. Therefore data aggregation is vital in order to reduce the communication overhead. However aggregator nodes can be a high risk to data confidentiality. To address the above challenge, the second layer of our protocol provides secure data aggregation by aggregating encrypted data without decryption at group leader nodes using homomorphic encryption. In our proposed solution we have provided different aggregation solutions according to the requirements of data confidentiality [62][117]. In our first solution data are decrypted at group leader nodes, aggregated, re-encrypted and sent towards the sink. The advantage in this scheme is that the group leaders can check the freshness and accuracy of the data received from their member sensor nodes before aggregating them. However in the event that a group leader or an aggregator node becomes compromised, this can compromise the data confidentiality of all member sensor nodes in the group. Therefore we have proposed the second method that allows data to be aggregated in an encrypted form to provide better data confidentiality. The requirement of secure data aggregation in any application depends upon its nature and data sensitivity. As we have described earlier our intention is to propose an integrated proactive security solution. Therefore the first and second layers are integrated with each other, allowing the two layers to provide better resilience against multiple threats. Furthermore we have tested the first and second layers together.

- In the third layer of our protocol we have proposed a novel secure group leader selection method which is part of our key management protocol. In our proposed method an old group leader sensor node will select a new group leader sensor node from its neighbouring sensor nodes based on a number of weighting factors. These weighting factors are calculated using the available energy at each candidate sensor node, its level of trust, its distance from the current group leader and the number of its neighbours. The sensor node with the highest calculated value will be selected as the new group leader sensor node. Unlike other group

leader selection schemes we do not involve all group member sensor nodes in the selection process, which significantly reduces the level of communication overhead. The energy cost of the selection process increases as the number of participant sensor nodes in the selection process increases.

- In the fourth layer of our protocol we have proposed a key management scheme for MSNs based on all the previous layers. As described earlier, mobility in WSNs brings additional strong security challenges. We have also defined a security policy for MSNs which has basic rules for mobile sensor nodes when roaming from one group to another. In this solution we have blended our solution with a probabilistic key management technique. We have used probabilistic key distribution only for mobile sensor node authentication when a sensor node roams from one group to another. The host group treats the guest mobile sensor node according to a given security policy. For our proposed protocol, all the layers are integrated with each other so as to provide a better proactive solution for static and mobile WSNs.

7.4 Future Work

So far in this chapter we have recapped the project aims, main findings and results, and considered the novel contributions of our work. For future research our current work can be extended in several directions. More functionality can be incorporated into the current layers, and also completely new layers with new security solutions could be added. As we have described earlier our protocol offers proactive security solutions. Therefore in the future work these solutions could be integrated with detective and reactive security solutions. The outcome of such integration will be a more effective security protocol for WSNs, where more layers can be added and removed according to the security requirements of different applications.

7.4.1 Secure Data Aggregation

Secure data aggregation is handled by the second layer of our protocol. We provide two different types of aggregation scheme, one with encrypted data and the other without. In the scheme where data are decrypted before aggregation at aggregator sensor nodes, it is possible to establish whether any junk data have been sent by a malicious sensor node. This scheme provides a benefit in this case, but as a consequence data confidentiality could be weakened in case an aggregator node becomes compromised. However, the scheme where data aggregation takes place without decryption provides a benefit in maintaining better data confidentiality, but the injection of random encrypted data by a malicious node can lead to incorrect aggregated results. Therefore in the future work the secure data aggregation scheme needs to be extended by adding functionality able to check the authenticity of data in its encrypted form. This will reduce the risk of incorrect results where a malicious sensor node exists in the WSN. However, increasing the security will inevitably increase resource consumptions. Therefore it is an important issue for the future work to investigate in order to balance these two concerns.

Furthermore in the future work it is important to investigate how effective the integration of our proactive solutions with reactive security solutions is in terms of tackling security attacks where malicious nodes authenticate themselves to a WSN and inject invalid data.

Finally it is also interesting to consider the use of mobile aggregator nodes for secure data aggregation.

7.4.2 Group Leader Selection

In our current work we have proposed a new group leader selection scheme which forms part of the third layer of our protocol. For the future work, further investigatory research relating to certain parameters is required. The first parameter to be investigated is the position of a group leader. In our current work we have only considered static WSNs and given high priorities to those sensor nodes which are closer to the existing (outgoing) group leader sensor node, in the selection process. However suppose that a selected group leader is a mobile sensor node able to roam within the group. This action can seriously

affect the performance of the group, especially in the case of a free roaming sensor node. Therefore a different method needs to be devised for the management of mobile sensor nodes acting as group leaders.

It will be very useful to investigate current proposed schemes for node trust values, which are an important parameter of our group leader selection scheme, to find whether current proposed schemes work efficiently with the SADI-GKM scheme, they do have any impacts on performance on SADI-GKM, or a new trust evaluation scheme should be proposed. Furthermore the future work needs to propose a solution to find that the values sent by member sensor nodes for group leader selection are accurate or not. For example an adversary or any selfish node may want to become a group leader so it can send false information to the current group leader.

Additionally in the future work it will be interesting to establish the performance of the current scheme in different types of WSNs and using different topologies.

7.4.3 Key Management for MSNs

An important challenge for the security of MSNs is how to make use of their limited resources to achieve scalability and anywhere security in a cost-effective manner. As described in Chapter 3, mobile sensor nodes may have different types of roaming behaviour. In the future research it will be interesting to investigate the relationships between different types of sensor node roaming and various security requirements. This research will help to identify important parameters needed for developing new security solutions for MSNs.

7.4.4 Future Security Models for WSNs

In the future work, more layers can be added into our protocol to provide security countermeasures against other security threats such as Black Hole, Sybil and many other routing attacks. Furthermore to create a best security model for WSNs in the future, it is important to integrate SADI-GKM with detective and reactive security solutions. During the development of security model, it is better to analysis the performance of current

reactive security schemes with SADI-GKM protocol together before integrating these components. There is a chance that current reactive solutions might not show good performance when combined with a proactive security scheme. This might happen in the case where a solution has been designed specifically for one problem space or application. In addition there should be a detective mechanism working along with the proactive part of the protocol. Whenever a threat is detected, the reactive security mechanism should become active. We believe such a self organised security model for WSNs will provide better security. However the research in this area is immature and there are difficult challenges remaining. We believe our current novel research contributions will help in the future development of secure WSNs.

7.5 Concluding Remarks

Recent advances in micro, electro and mechanical systems technologies, wireless communication and digital electronics have enabled the development of low cost, low power and multifunctional sensor nodes that are small in size and communicate over short distances. These tiny sensor nodes, which consist of data sensing, processing and communication components, leverage the idea of WSNs. The reason for WSN's popularity is due in part to the small sizes and low costs of sensors, their operations and the networking behaviours, which enable them to provide significant advantages for many applications that would not have been possible in the past.

Alongside energy efficient communication protocols we require a balanced security solution guarding against possible security threats in WSNs. It's interesting to note that WSNs face not only the same security challenges as traditional networks (LAN, WAN, MAN and etc.) but also additional difficulties in the limited resources of sensor nodes. As a result, we are unable to use traditional techniques for WSNs. A challenging and distinct security problem in WSNs is node capture attacks, where an adversary gains full control over a sensor node through direct physical access to it. This can lead to a compromise in the communication of the entire WSN. The compromised sensor node can be an aggregator node, a cluster head node or a normal sensor node. Therefore we should consider such threats as a high risk to communication and data confidentiality/security.

Furthermore in case a group leader sensor node suffers complete resource depletion, or is compromised, the selection of a new group leader in a secure way is vital.

In contrast to all existing work, we have proposed a novel protocol SADI-GKM for structure and density independent group based key management for large-scale WSNs. This protocol is designed to provide multiple security services such as improved secure communication in static and mobile WSNs, secure data aggregation, data confidentiality, secure group leader selection, resilience against node capture and replication attacks, and protection against malicious nodes from sending random encrypted data in an aggregated form. SADI-GKM provides these services using reduced memory and processing overheads as well as high connectivity as compared to existing schemes. In particular, the structure and density independence helps to reduce computation costs in maintaining the network topology when new nodes join or leave a group.

We have analysed and evaluated the proposed protocol using simulation techniques. Our evaluation was focused on resilience against node capture and replay attacks, power consumption reduction through the use of secure data aggregation, energy cost reduction for secure group leader selection, and energy cost reduction for the key management of MSNs. By comparing our results to those of other solutions available in the literature, our work provides better resilience against node capture attacks, replication attack, efficient secure data aggregation for data confidentiality, energy efficient and group leader selection scheme. The experiments have shown that our protocol is scalable and structure independent.

SADI-GKM can be immediately and efficiently used in indoor applications for example monitoring building and factories for fire, pipe leakage and hidden moisture and also in health sector. In out door applications, it can be used in traffic monitoring and monitoring entire city for better fire rescue systems. In future application its can be used in application to monitor forest and large group areas but efficient deployment mechanism is required before implementation.

Research in WSNs, especially in the WSN security, is still immature. There are still many research challenges to be addressed in order to implement WSNs realistically in our daily

life. We believe that our novel protocol SADI-GKM and investigatory research findings will help toward the future secure development of WSNs.

Appendix

Publications

Book Chapter

1. Kashif Kifayat, Madjid Merabti, Qi Shi, David Llewellyn-Jones, "Security in Wireless Sensor Networks", Hand Book on Communication and Information Security to be published by Springer, in press.

Articles in Refereed Journals

2. Kashif Kifayat, Madjid Merabti, Qi Shi, David Llewellyn-Jones, "Application Independent Dynamic Group-Based Key Establishment for Large-scale Wireless Sensor Networks", China Communications Journal, vol. 4 (1), pp. 14-27, February 2007. China Institute of Communications.

3. Kashif Kifayat, Madjid Merabti, Qi Shi, David Llewellyn-Jones, "Group-Based Secure Communication for Large Scale Wireless Sensor Networks", Journal of Information Assurance and Security (JIAS), vol. 2 (2), June 2007.

Articles in Refereed Conferences and Workshops

4. Kashif Kifayat, Madjid Merabti, Qi Shi, David Llewellyn-Jones, "Applying Secure Data Aggregation Techniques for a Structure and Density Independent Group Based Key Management Protocol", The Third IEEE International Symposium on Information Assurance and Security (IAS 2007), Manchester, UK, 29-31 August 2007

5. Kashif Kifayat, Madjid Merabti, Qi Shi, David Llewellyn-Jones, "The Performance of Dynamic Group-Based Key Establishment (DGKE) under Node Capture Attacks in Wireless Sensor Networks", 2nd Conference on Advances in Computer Security and Forensics (ACSF 2007), Liverpool, UK, 12-13 July 2007.

6. Kashif Kifayat, Madjid Merabti, Qi Shi, David Llewellyn-Jones, "Modelling Node Capture Attacks in Wireless Sensor Networks Using Simulation", 23rd Annual UK Performance Engineering Workshop, Edge Hill University, UK, 9-10 July 2007.

7. Kashif Kifayat, Madjid Merabti, Qi Shi, David Llewellyn-Jones, "Security in Mobile Wireless Sensor Networks", 8th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet 2007), Liverpool, UK, 28-29 June 2007.

8. Kashif Kifayat, Madjid Merabti, Qi Shi, David Llewellyn-Jones, "Dynamic Group-Based Key Establishment for Large-scale Wireless Sensor Networks".

First International Conference on Communications and Networking in China (Chinacom 2006), Beijing, China, 25-27 October 2006. (Invited).

9. Kashif Kifayat, Madjid Merabti, Qi Shi, David Llewellyn-Jones, "Key Management for Wireless Sensor Networks in Building Environments". PGNet 2006, Liverpool, UK, 26-27 June 2006.

10. Kashif Kifayat, Madjid Merabti, Qi Shi, David Llewellyn-Jones, "Topology creation and adaptive routing for wireless sensor network in building environment", PGNet 2005, Liverpool, UK, June 2005.

References

[1]. Callaway, H., "Wireless Sensor Networks: Architectures and Protocols", 0849318238, 9780849318238, CRC Press, 2003.

[2]. FunctionX. "Computer Networks". Available at: http://www.functionx.com/networking/Lesson01.htm, Access date:9/8/2005.

[3]. Nicopolitidis, P. "Wireless Networks", Vol. 2, 0470845295: John Wiley and Sons. 2003.

[4]. Raghavendra, C.S., K.M. Sivalingam, and T. Znati. "Wireless Sensor Networks", 978-1-4020-7883-5: Springer US. 2004.

[5]. Xu, Y., J. Heidemann, and D. Estrin. "Geography-informed Energy Conservation for Ad Hoc Routing". in Mobicom 2001, pp. 70-84, 2001.

[6]. Piotrowski, K., P. Langendoerfer, and S. Peter. "How Public Key Cryptography Influences Wireless Sensor Node Lifetime". in 4th ACM Workshop on Security of ad hoc and Sensor Networks (SASN) 2006, pp. 169 – 176, 2006.

[7]. Akyildiz, I.F., Y.S. W. Su, and E. Cayirci. "A Survey on Sensor Networks", IEEE Communications Magazine, Vol. 40, Iss. 8, pp. 102-116, August 2002.

[8]. He, T., S. Krishnamurthy, J.A. Stankovic, T. Abdelzaher, R.S. L. Luo, T. Yan, L. Gu, G. Zhou, J. Hui, and B. Krogh. "VigilNet: An Integrated Sensor Network System for Energy-Efficient Surveillance", Vol. 2, Iss. 1, pp. 1-38, February 2006.

[9]. Stankovic, J. "Wireless Sensor Networks": Chapter in Handbook of Real-Time and Embedded Systems, CRC Press. to appear.

[10]. "VigilNet". Available at: http://www.cs.virginia.edu/wsn/vigilnet/ Access date:31/12/2006.

[11]. Cerpa, A., J. Elson, M. Hamilton, and J. Zhao. "Habitat Monitoring: Application Driver for Wireless Communications Technology". in ACM SIGCOMM 2001. Costa Rica, pp. 20-41, April 2001.

[12]. Halweil, B. "Study finds modern farming is costly", World Watch 14, Iss., pp. 910, 2001.

[13]. Biagioni, E. and K. Bridges. "The Application of Remote Sensor Technology to Assist the Recovery of Rare and Endangered Species", Special issue on Distributed Sensor Networks for the International Journal of High Performance Computing Applications, Vol. 16, Iss. 3, pp. 00-00, August 2002.

[14]. "ALERT". Available at: http://www.alertsystems.org, Access date:01/01/2007.

[15]. Patel, S., K. Lorincz, R. Hughes, N. Huggins, J.H. Growdon, M. Welsh, and P. Bonato. "Analysis of Feature Space for Monitoring Persons with Parkinson's Disease With Application to a Wireless Wearable Sensor System". in 29th IEEE EMBS Annual International Conference 2007. Lyon, France, pp. 6290-6293, August 2007.

[16]. Gao, T., T. Massey, L. Selavo, M. Welsh, and M. Sarrafzadeh. "Participatory User Centered Design Techniques for a Large Scale Ad-Hoc Health Information System", In First International Workshop on Systems and Networking Support for Healthcare and Assisted Living Environments (HealthNet) 2007. San Juan, Puerto Rico, pp. 43 – 48, June 2007.

[17]. Gao, T., D. Greenspan, M. Welsh, R.R. Juang, and A. Alm. "Vital Signs Monitoring and Patient Tracking Over a Wireless Network". in 27th IEEE EMBS Annual International Conference 2005, pp. 102-105, September 2005.

[18]. Gao, T., D. Greenspan, and M. Welsh. "Improving Patient Monitoring and Tracking in Emergency Response". in International Conference on Information Communication Technologies in Health 2005, pp. July 2005.

[19]. Malan, D., T.R.F. Fulford-Jones, M. Welsh, and S. Moulton. "CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care". in MobiSys Workshop on Applications of Mobile Embedded Systems (WAMES) 2004. Boston, MA, pp. 12-14, June 2004.

[20]. Noury, N., T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, and T. Porcheron. "Monitoring behaviour in home using a smart fall sensor". in IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology 2000, pp. 607–610, October 2000.

[21]. Cerpa, A. and D. Estrin. "ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies", Mobile Computing, IEEE Transactions, Vol. 3, Iss. 3, pp. 272-285, July 2004.

[22]. Song, H., S. Zhu, and G. Cao. "SVATS: A Sensor-network-based Vehicle Anti-Theft System". in IEEE INFOCOM 2008, pp. 2128-2136, 13-18 April 2008.

[23]. Rabaey, J., J. Ammer, J.L.d.S. Jr., and D. Patel. "Pico-Radio: Ad-Hoc Wireless Networking of Ubiquitous Low Energy Sensor/Monitor Nodes". in IEEE Computer Society Annual Workshop on VLSI (WVLSI) 2000. Orlanda, Florida, pp. 9–12, April 2000.

[24]. Jovanov, E., A. Milenkovic, C. Otto, and P.C.d. Groen. "A Wireless Body Area Network of Intelligent Motion Sensors for Computer Assisted Physical Rehabilitation", Journal of Neuro Engineering and Rehabilitation, Iss., pp. 18-29, March 2005.

[25]. Welch, J., F. Guilak, and S.D. Baker. "A Wireless ECG Smart Sensor for Broad Application in Life Threatening Event Detection". in 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society 2004. San Francisco, CA, pp. 3447-3449, September 2004.

[26]. Milenkovic, A., C. Otto, and E. Jovanov. "Wireless Sensor Networks for Personal Health Monitoring: Issues and an Implementation", Computer Communications, Special issue: Wireless Sensor Networks: Performance, Reliability, Security, and Beyond, Vol. 29, Iss. 13 and 14, pp. 2521-2533, 2006.

[27]. Demirbas, M. *Scalable design of fault-tolerance for Wireless Sensor Networks*. 2004, the Ohio State University.

[28]. Koushanfar, F., M. Potkonjak, and A. Sangiovanni-Vincentelli. "Fault-Tolerance in Sensor Networks". Handbook of Sensor Networks, ed. I. Mahgoub and M. Ilyas, Vol. 36: CRC press. 2004.

[29]. Krishnan, M. *Intrusion Detection in Wireless Sensor Networks*. 2006, Department of EECS, University of California at Berkeley.

[30]. Pottie, G.J. and W.J. Kaiser. "Wireless Integrated Network Sensors", Communications of the ACM, Vol. 43, Iss. 5, pp. 551–558, May 2000.

[31]. Kahn, J.M., R.H. Katz, and K.S.J. Pister. "Next Century Challenges: Mobile Networking For Smart Dust". in ACM MobiCom 1999. Washington, USA, pp. 271–278, 1999.

[32]. Shih, E., S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan. "Physical layer driven protocol and algorithm design for energy-efficient Wireless Sensor Networks". in ACM MobiCom 2001. Rome, Italy, pp. 272–286, July 2001.

[33]. Kifayat, K., M. Merabti, Q. Shi, and D. Llewellyn-Jones. "The Performance of Dynamic Group-Based Key Establishment (DGKE) under Node Capture Attacks in Wireless Sensor Networks". in 2nd Conference on Advances in Computer Security and Forensics (ACSF) 2007. Liverpool, UK, pp. 12-13 July 2007.

[34]. Bulusu, N., D. Estrin, L. Girod, and J. Heidemann. "Scalable coordination for Wireless Sensor Networks: self- configuring localization systems". in International Symposium on Communication Theory and Applications 2001. Ambleside, UK, pp. July 2001.

[35]. Al-Karaki, J.N. and A.E. Kamal. "Routing Techniques in Wireless Sensor Networks: A Survey", IEEE Wireless Communications, Vol. 11, Iss. 6, pp. 6- 28, 2004.

[36]. Kifayat, K., M. Merabti, Q. Shi, and D. Llewellyn-Jones. "Security in Mobile Wireless Sensor Networks". in 8th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet) 2007. Liverpool, UK, pp. 28-29 June 2007.

[37]. Ye, F., H. Luo, J. Cheng, S. Lu, and L. Zhang. "A Two-Tier Data Dissemination Model for Large-Scale Wireless Sensor Networks". in ACM/IEEE MOBICOM 2002, pp. 148 – 159, 2002.

[38]. Carman, D.W., P.S. Krus, and B.J. Matt. "Constraints and Approaches for Distributed Sensor Network Security", Technical Report 00-010, NAI Labs, Network Associates, Inc, Glenwood, MD, Iss., 2000.

[39]. Perrig, A., R. Szewczyk, J.D. Tygar, V. Wen, and D.E. Culler. "Spins: Security Protocols For Sensor Networks", Wireless Networking, Vol. 8, Iss. 5, pp. 521–534, 2002.

[40]. Shi, E. and A. Perrig. "Designing Secure Sensor Networks", Wireless Communication Magazine, Vol. 11, Iss. 6, pp. 38- 43, December 2004.

[41]. Walters, J.P., Z. Liang, W. Shi, and V. Chaudhary. "Wireless Sensor Networks Security: A Survey", in *Security in Distributed, Grid, and Pervasive Computing*, Y. Xiao, Editor. 2007, Aurebach Publications. p. 367-410.

[42]. Karygiannis, T. and L. Owens. *Wireless Network Security: 802.11, Bluetooth, and Handheld Devices*. 2002, NIST.

[43]. Liu, D., P. Ning, S. Zhu, and S. Jajodia. "Practical Broadcast Authentication in Sensor Networks". in 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous) 2005, pp. 118-129, July 2005.

[44]. Liu, D. and P. Ning. "Efficient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks". in 10th Annual Network and Distributed System Security Symposium 2003, pp. 263–276, February 2003.

[45]. Liu, D. and P. Ning. "Multilevel μTesla: Broadcast Authentication for Distributed Sensor Networks", Transactions in Embedded Computing System, Vol. 3, Iss. 4, pp. 800-836, 2004.

[46]. Engelbrecht, N. and W.T. Penzhorn. "Secure Authentication Protocols Used for Low Power Wireless Sensor Networks". in IEEE International Symposiumon 2005, pp. 1777-1782, 20-23 June 2005.

[47]. Wong, K.H.M., Y. Zheng, J. Cao, and S. Wang. "A Dynamic User Authentication Scheme for Wireless Sensor Networks". in IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC) 2006, pp. 244-251, Jun 2006.

[48]. Benenson, Z., N. Gedicke, and O. Raivio. "Realizing Robust User Authentication in Sensor Networks". in Workshop on Real-World Wireless Sensor Networks (REALWSN) 2005, pp. 20-21 June 2005.

[49]. Gruteser, M., G. Schelle, A. Jain, R. Han, and D. Grunwald. "Privacy-Aware Location Sensor Networks". in 9th conference on Hot Topics in Operating Systems 2003, pp. 28 – 28, 2003.

[50]. Langheinrich, M. "A Privacy Awareness System for Ubiquitous Computing Environments". in 4th International Conference on Ubiquitous Computing 2002, pp. 2002.

[51]. Snekkenes, E. "Concepts for Personal Location Privacy Policies". in 3rd ACM conference on Electronic Commerce 2001: ACM Press, pp. 48-57, 2001.

[52]. Duri, S., M. Gruteser, X. Liu, P. Moskowitz, R. Perez, M. Singh, and J. Tang. "Framework for Security And Privacy In Automotive Telematics". in 2nd ACM International Workshop on Mobile Commerce 2002, pp. 25-32, 2002.

[53]. Sweeney, L. "Achieving k-Anonymity Privacy Protection Using Generalization and Suppression", International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, Vol. 10, Iss. 5, pp. 571-588, 2002.

[54]. Karlof, C. and D. Wagner. "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", Elsevier's Ad Hoc Networks Journal, Special Issue on Sensor Network Applications and Protocols, Vol. 1, Iss. 1, pp. 293-315, September 2003.

[55]. Alzaid, H., E. Foo, and J.M.G. Nieto. "Secure data aggregation in wireless sensor network: a survey". in Proceedings of the sixth Australasian conference on Information security 2008. Wollongong, NSW, Australia, pp. 93-105, 2008.

[56]. Chan, H., A. Perrig, and D. Song. "Secure Hierarchical In-Network Aggregation in Sensor Networks". in 13th ACM conference on Computer and communications security 2006. Alexandria, Virginia, USA, pp. 278 – 287, 2006.

[57]. Intanagonwiwat, C., D. Estrin, R. Govindan, and J. Heidemann. "Impact of Network Density on Data Aggregation in Wireless Sensor Networks". in 22nd International Conference on Distributed Computing Systems 2002, pp. 457, 2002.

[58]. Madden, S., M.J. Franklin, J.M. Hellerstein, and W. Hong. "TAG: a Tiny Aggregation Service For Ad-Hoc Sensor Networks", SIGOPS Operating System Review, Vol. 36, Iss. SI, pp. 131–146, 2002.

[59]. Yao, Y. and J. Gehrke. "The COUGAR Approach to In-Network Query Processing In Sensor Networks", SIGMOD Record, Vol. 31, Iss. 3, pp. 9-18, 2002.

[60]. Deshpande, A., S. Nath, P.B. Gibbons, and S. Seshan. "Cache-and-Query for Wide Area Sensor Databases". in International Conference on Management of Data SIGMOD 2003. San Diego, California, pp. 503-514, 2003.

[61]. Castelluccia, C., E. Mykletun, and G. Tsudik. "Efficient Aggregation of Encrypted Data in Wireless Sensor Networks". in 2nd Annual International Conference on Mobile and Ubiquitous Systems 2005, pp. 109- 117, 17-21 July 2005.

[62]. Kifayat, K., M. Merabti, Q. Shi, and D. Llewellyn-Jones. "Applying Secure Data Aggregation Techniques for a Structure and Density Independent Group Based Key Management Protocol". in Third IEEE International Symposium on Information Assurance and Security (IAS) 2007. Manchester, UK, pp. 44-49, 29-31 August 2007.

[63]. Saxena, M. *Security in Wireless Sensor Networks - A Layer based Classification.* CERIAS Technical Report, 2007.

[64]. "IDs". Available at: http://searchsecurity.techtarget.com, Access date:10/12/2006.

[65]. Freiling, F., I. Krontiris, and T. Dimitriou. "Towards Intrusion Detection in Wireless Sensor Networks". in 13th European Wireless Conference 2007. Paris, France, pp. April 2007.

[66]. Krebs, B. "A Short History of Computer Viruses and Attacks". Available at: http://www.washingtonpost.com/ac2/wp-dyn/A50636-2002Jun26, Access date:12/10/2007.

[67]. "The History of Computer Viruses". Available at: http://www.virus-scan-software.com/virus-scan-help/answers/the-history-of-computer-viruses.shtml, Access date:12/10/2007.

[68]. Innella, P. "A Brief History of Network Security and the Need for Host Based Intrusion Detection". Available at: http://www.tdisecurity.com/resources/, Access date:01/10/2007.

[69]. Becher, A., Z. Benenson, and M. Dornseif. "Tampering with Motes: Real-world Physical Attacks on Wireless Sensor Networks". in International Conference on Security in Pervasive Computing (SPC) 2006, pp. 104-118, April 2006.

[70]. Kocher, P., J. Jaffe, and B. Jun. "Differential Power Analysis". in Advances in Cryptography – CRYPTO'99, Lecture Notes in Computer Science 1999: Springer-Verlag pp. 388-397, 1999.

[71]. Kifayat, K., M. Merabti, Q. Shi, and D. Llewellyn-Jones. "Application Independent Dynamic Group-Based Key Establishment for Large-scale Wireless Sensor Networks", China Communications, Vol. 4, Iss. 1, pp. 14-27, February 2007.

[72]. Eschenauer, L. and V. Gligor. "A Key Management Scheme for Distributed Sensor Networks". in 9th ACM Conference on Computer and Communication Security 2002, pp. 41 - 47, 2002.

[73]. Hac, A. "Wireless Sensor Networks", 0-470-86736-1: John Wiley & Sons. 2003.

[74]. Liu, D., P. Ning, and W. Du. "Group-Based Key Pre-Distribution in Wireless Sensor Networks". in ACM Workshop on Wireless Security (WiSe) 2005, pp. 11 – 20, September 2005.

[75]. Chan, H., 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, 2003.

[76]. Chan, H. and A. Perrig. "PIKE: Peer Intermediaries for Key Establishment in Sensor Networks". in INFOCOM 2005, pp. 524- 535, 2005.

[77]. Eltoweissy, M., A. Wadaa, S. Olariu, and L. Wilson. "Group Key Management Scheme for Large-Scale Sensor Networks", Elsevier Ad Hoc Networks Vol. 3, Iss. 5, pp. 668-688, 2004.

[78]. Du, W., J. Deng, Y.S. Han, S. Chen, and P. Varshney. "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge". in IEEE INFOCOM 2004, pp. March 2004.

[79]. Huang, D., M. Mehta, D. Medhi, and L. Harn. "Location-Aware Key Management Scheme for Wireless Sensor Networks". in 2nd ACM Workshop on Security of ad hoc and sensor networks (SASN) 2004, pp. 29-42, October 2004.

[80]. Zhou, L., J. Ni, and C.V. Ravishankar. "Efficient Key Establishment for Group-Based Wireless Sensor Deployments". in WiSe 2005, pp. 1 - 10, 2 September 2005.

[81]. Roosta, T., S. Shieh, and S. Sastry. "Taxonomy of Security Attacks in Sensor Networks". in First IEEE International Conference on System Integration and Reliability Improvements 2006. Hanoi, Vietnam, pp. 13-15 December 2006.

[82]. Okeya, K. and T. Iwata. "Side Channel Attacks on Message Authentication Codes", IPSJ Digital Courier, Vol. 2, Iss., pp. 478-488, 2006.

[83]. Tiu, C.C. *A New Frequency-Based Side Channel Attack for Embedded Systems*. 2005.

[84]. Agrawal, D., B. Archambeault, J.R. Rao, and P. Rohatgi. *The EM Side-Channel (s): Attacks and Assessment Methodologies*. 2002, IBM.

[85]. Karlof, C., N. Sastry, and D. Wagner. "Tinysec: A Link Layer Security Architecture for Wireless Sensor Networks". in 2nd ACM Conference on Embedded Networked Sensor Systems 2004, pp. 162 – 175, November 2004.

[86]. Perrig, A., D. Wagner, and J. Stankovic. "Security in Wireless Sensor Networks", Communications of the ACM, Vol. 47, Iss. 6, pp. 53 - 57, June 2004.

[87]. Wood, A.D. and J.A. Stankovic. "Denial of Service in Sensor Networks", IEEE Computer, Vol. 35, Iss. 10, pp. 54-62, October 2002.

[88]. Wood, A.D. and J.A. Stankovic. "A Taxonomy for Denial-of-Service Attacks in Wireless Sensor Networks", Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, Iss.,

[89]. "Volcano monitoring". Available at: http://www.eecs.harvard.edu/~mdw/proj/volcano/ Access date:10/1/2007.

[90]. Werner-Allen, G., K. Lorincz, J. Johnson, J. Lees, and M. Welsh. "Fidelity and Yield in a Volcano Monitoring Sensor Network". in 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI) 2006. Seattle, pp. 381 – 396, November 2006.

[91]. Werner-Allen, G., K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh. "Deploying a Wireless Sensor Network on an Active Volcano, Special Issue on Data-Driven Applications in Sensor Networks", IEEE Internet Computing, Vol. 10, Iss. 2, pp. 18- 25, March-April 2006.

[92]. Wood, A., G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, and J. Stankovic. *ALARM-NET: Wireless Sensor Networks for Assisted-Living and Residential Monitoring*. 2006, Department of Computer Science, University of Virginia.

[93]. Virone, G., T. Doan, A. Wood, and J.A. Stankovic. "Dynamic Privacy in Assisted Living and Home Health Care". in Joint Workshop On High Confidence Medical Devices, Software, and Systems (HCMDSS) and Medical Device Plug-and-Play (MD PnP) Interoperability 2007, pp. 2007.

[94]. "AlarmNet". Available at: http://www.cs.virginia.edu/wsn/medical/, Access date:10/1/2007.

[95]. Juang, P., H. Oki, Y. Wang, M. Martonosi, LiShiuan, Peh, and D. Rubenstein. "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet". in 10th international conference on Architectural support for programming languages and operating systems 2002. San Jose, CA, pp. 96 – 107, October 2002.

[96]. "ZebraNet". Available at: http://www.princeton.edu/~mrm/zebranet.html, Access date:3/11/2006.

[97]. Regehr, J., N. Cooprider, W. Archer, and E. Eide. *Memory Safety and Untrusted Extensions for Tinyos*. 2006, School of Computing, University of Utah.

[98]. Cooprider, N., W. Archer, E. Eide, D. Gay, and J. Regehr. "Efficient Memory Safety for TinyOS". in 5th ACM Conference on Embedded Networked Sensor Systems (SenSys) 2007. Sydney, Australia, pp. 205 – 218, November 2007.

[99]. Hill, J., P. Bounadonna, and D. Culler. *Active Message Communication for Tiny Network Sensors*. 2001, UC Berkeley ,Berkeley.

[100]. Shaneck, M., K. Mahadevan, V. Kher, and Y. Kim. "Remote Software-Based Attestation for Wireless Sensors". in Second European Workshop on Security and Privacy in Ad Hoc and Sensor Networks 2005, pp. 27- 41, July 2005.

[101]. Perrig, A., J. Newsome, E. Shi, and D. Song. "The Sybil Attack In Sensor Networks: Analysis And Defences". in 3ard International Symposium on Information Processing in Sensor Networks 2004: ACM Press, pp. 259–268, 2004.

[102]. Raffo, D. *Security Schemes for the OLSR Protocol for Ad Hoc Networks, Chapter 3*. 2005, University of Paris.

[103]. Hu, Y.-C., A. Perrig, and D.B. Johnson. "Adriane: A Secure On-Demand Routing Protocol for Ad Hoc Networks". in Annual ACM International Conference on Mobile Computing and Networking (MobiCom) 2002, pp. September 2002.

[104]. Hu, Y.-C., A. Perrig, and D.B. Johnson. "Packet Leashes: A Defence Against Wormhole Attacks In Wireless Ad Hoc Networks". in 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) 2003. San Francisco, CA, pp. 1976- 1986, April 2003.

[105]. Yu, B. and B. Xiao. "Detecting Selective Forwarding Attacks in Wireless Sensor Networks". in 20th International Parallel and Distributed Processing Symposium IPDPS 2006. Greece, pp. 1-8, 25-29 April 2006.

[106]. Pirzada, A.A. and C. McDonald. "Circumventing Sinkholes and Wormholes in Wireless Sensor Networks". in International Workshop on Wireless Ad-hoc Networks 2005, pp. 2005.

[107]. Datema, S. *A Case Study of Wireless Sensor Network Attacks*. 2005.

[108]. Hartung, C., S. Holbrook, R. Han, and C. Seielstad. "FireWxNet: A Multi-Tiered Portable Wireless System for Monitoring Weather Conditions in Wildland Fire Environments". in Fourth International Conference on Mobile Systems, Applications and Services (MobiSys), 2006, pp. 28-41, 2006.

[109]. Deng, J., R. Han, and S. Mishra. "Countermeasures Against Traffic Analysis Attacks in Wireless Sensor Networks". in first IEEE/Cerate Net Conference on Security and Privacy in Communication Networks (SecureComm) 2005. Athens, Greece, pp. 113-124, September 2005.

[110]. "Xbow". Available at: http://www.xbow.com/wirelesshome.aspx, Access date:12/07/2007.

[111]. Rowe, A., R. Mangharam, and R. Rajkumar. "RT-Link: A Global Time-Synchronized Link Protocol for Sensor Networks", Elsevier Ad hoc Networks, Special Issue on Energy efficient design in wireless ad hoc and sensor networks Iss.,

[112]. Farrugia, E. and R. Simon. "An Efficient and Secure Protocol for Sensor Network Time Synchronization", Journal of Systems and Software, Vol. 79, Iss. 2, pp. 147-162, February 2006.

[113]. Wu, S. and K.S. Candan. "Demand-Scalable Geographic Multicasting In Wireless Sensor Networks", Computer Communications, Vol. 33, Iss. 14-15, pp. 2931-2953, 15 October 2007.

[114]. Wang, Y., J.Gao, and J.S.B. Mitchell. "Boundary Recognition in Sensor Networks by Topological Methods". in 12th Annual International Conference on Mobile Computing and Networking (MobiCom) 2006, pp. 122-133, September, 2006.

[115]. "RSA". Available at: http://www.rsasecurity.com/rsalabs/, Access date:2/2/2006.

[116]. Saraogi, M. *Security in Wireless Sensor Networks*. 2005, Department of Computer Science University of Tennessee, Knoxville.

[117]. Kifayat, K., M. Merabti, Q. Shi, and D. Llewellyn-Jones. "Group-Based Secure Communication for Large Scale Wireless Sensor Networks", Journal of Information Assurance and Security (JIAS), Iss.,

[118]. Kifayat, K., M. Merabti, Q. Shi, and D. Llewellyn-Jones. "Modelling Node Capture Attacks in Wireless Sensor Networks Using Simulation". in 23rd Annual UK Performance Engineering Workshop 2007. Edge Hill University, UK, pp. 9-10 July 2007.

[119]. Ahmed, I. *Efficient Key Management for Wireless Sensor Networks*, in *Computer Science*. 2006, Liverpool Hope University: Liverpool.

[120]. Muraleedharan, R. and L.A. Osadciw. "Cross Layer Protocols in Wireless Sensor Networks (Poster)". in IEEE Infocomm Student Workshop 2006, pp. April 2006.

[121]. Zhu, S., S. Setia, and S. Jajodia. "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks". in ACM Conference on Computer and Communications Security (CCS) 2003. Washington D.C, pp. 62- 72, October 2003.

[122]. Du, X., Y. Xiao, M. Guizani, and H.H. Chen. "An Effective Key Management Scheme for Heterogeneous Sensor Networks", Elsevier Ad Hoc Networks Vol. 5, Iss. 1, pp. 24–34, January 2007.

[123]. Du, X., M. Guizani, Y. Xiao, S. Ci, and H.H. Chen. "A Routing-Driven Elliptic Curve Cryptography Based Key Management Scheme for Heterogeneous Sensor Networks". in IEEE Transactions on Wireless Communications (ICC) 2007, pp. 3407-3412, 2007.

[124]. Traynor, P., H. Choi, G. Cao, S. Zhu, and T. Porta. "Establishing Pair-Wise Keys In Heterogeneous Sensor Networks". in IEEE INFOCOM 2006, pp. 1-12, 2006.

[125]. Ning, P., R. Li, and D. Liu. "Establishing Pairwise Keys In Distributed Sensor Networks", ACM Transactions on Information and System Security, Vol. 8, Iss. 1, pp. 41-77, 2005.

[126]. Anjum, F. "Location Dependent Key Management Using Random Key Predistribution in Sensor Networks". in WiSe 2006, pp. 21 – 30, 2006.

[127]. Younis, M.F., K. Ghumman, and M. Eltoweissy. "Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Networks", IEEE Transactions on Parallel and Distributed Systems, Vol. 17, Iss. 8, pp. 865–882, 2006.

[128]. Liu, D. and P. Ning. "Establishing Pairwise Keys in Distributed Sensor Networks". in 10th ACM Conference on Computer and Communications Security (CCS) 2003. Washington D.C, pp. 52-61, October 2003.

[129]. Lai, B.C., S. Kim, and I. Verbauwhede. "Scalable Session Key Construction Protocol For Wireless Sensor Networks". in IEEE Workshop on Large Scale Real-Time and Embedded Systems (LARTES) 2002, pp. December 2002.

[130]. Pietro, R.D., L.V. Mancini, and S. Jajodia. "Providing Secrecy In Key Management Protocols For Large Wireless Sensors Networks", Ad Hoc Networks, Vol. 1, Iss. 4, pp. 455-468, November 2003.

[131]. Jolly, G., M.C. Kuscu, P. Kokate, and M. Younis. "A Low-Energy Key Management Protocol for Wireless Sensor Networks". in IEEE Symposium on Computers and Communications (ISCC) 2003. Kemer - Antalya, Turkey, pp. 335- 340, 30 June- 3 July 2003.

[132]. Bohge, M. and W. Trappe. "An Authentication Framework for Hierarchical Ad Hoc Sensor Networks". in 2nd ACM workshop on Wireless security (WiSe) 2003, pp. 79 – 87, 2003.

[133]. Fu, H., S. Kawamura, M. Zhang, and L. Zhang. "Replication Attack on Random Key Pre-Distribution Schemes for Wireless Sensor Networks", Computer Communications, Vol. 31, Iss. 4, pp. 842-857, July 2008.

[134]. Fu, H., S. Kawamura, M. Zhang, and L. Zhang. "Replication Attack on Random Key Pre-Distribution Schemes for Wireless Sensor Networks". in IEEE Workshop on Information Assurance and Security 2005. US Military Academy, West Point, NY, pp. 134- 141, 2005.

[135]. Du, W., J. Deng, Y. Han, and P. Varshney. "A Pairwise Key Pre-Distribution Scheme For Wireless Sensor Networks". in 10th ACM Conference on Computer and Communications Security (CCS) 2003. Washington, DC, USA, pp. 42–51, 2003.

[136]. Du, W., J. Deng, Y. Han, P. Varshney, J. Katz, and A. Khalili. "A Pairwise Key Predistribution Scheme For Wireless Sensor Networks", ACM Transactions on Information and System Security, Vol. 8, Iss. 2, pp. 228–258, 2005.

[137]. Parno, B., A. Perrig, and V. Gligor. "Distributed Detection of Node Replication Attacks in Sensor Networks". in IEEE Symposium on Security and Privacy 2005. Oakland, CA, pp. 49 – 63, May 2005.

[138]. Traynor, P., R. Kumar, H.B. Saad, G. Cao, and T.L. Porta. *Liger: Implementing Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks*. 2005, Penn State University.

[139]. Heinzelman, W.R., A. Chandrakasan, and H. Balakrishnan. "Energy Efficient Communication Protocol for Wireless Microsensor Networks". in Hawaiian International Conference On Systems Science 2000, pp. January 2000.

[140]. Wen, C.Y. and W.A. Sethares. "Adaptive Decentralized Re-Clustering For Wireless Sensor Networks". in SMC 2006. Taipei, Taiwan, pp. 2709-2716, 2006.

[141]. Wu, B., J. Wu, E.B. Fernandez, M. Ilyas, and S. Magliveras. "Secure And Efficient Key Management In Mobile Ad Hoc Networks", Journal of Network and Computer Applications, Vol. 30, Iss. 3, pp. 937-954, August 2007.

[142]. Cho, J., I. Chen, and D. Wang. "Performance Optimization of Region-Based Group Key Management in Mobile Ad Hoc Networks", Performance Evaluation, Vol. 65, Iss. 5, pp. 319-344, 26 July 2007.

[143]. Wang, N. and S. Fang. "A Hierarchical Key Management Scheme For Secure Group Communications In Mobile Ad Hoc Networks", Journal of Systems and Software, Vol. 80, Iss. 10, pp. 1667-1677, October 2007.

[144]. Xiao, Y., V.K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway. "A Survey of Key Management Schemes in Wireless Sensor Networks", Computer Communications, Vol. 30, Iss. 11-12, pp. 2314-2341, 10 September 2007.

[145]. Hussain, S., F. Kausar, and A. Masood. "An Efficient Key Distribution Scheme for Heterogeneous Sensor Networks". in ACM International Conference on Wireless communications and Mobile Computing (IWCMC) 2007, pp. 388 – 392, August 2007.

[146]. Vasudevan, S., B. DeCleene, N. Immerman, J.F. Kurose, and D.F. Towsley. "Leader Election Algorithms for Wireless Ad Hoc Networks". in DISCEX 2003, pp. 261-272, 2003.

[147]. Cam, H., S. Ozdemir, P. Nair, D. Muthuavinashiappan, and H.O. Sanli. "Energy-Efficient Secure Pattern Based Data Aggregation For Wireless Sensor Networks", Computer Communications, Vol. 29, Iss., pp. 446–455, 2006.

[148]. Jadia, P. and A. Mathuria. "Efficient Secure Aggregation In Sensor Networks". in 11th International Conference on High Performance Computing 2004, pp. 2004.

[149]. Przydatek, B., D. Song, and A. Perrig. "SIA: Secure Information Aggregation In Sensor Networks". in 1st International Conference on Embedded Networked Sensor Systems 2003, pp. 255 – 265, 2003.

[150]. Hu, L. and D. Evans. "Secure Aggregation for Wireless Networks". in Workshop on Security and Assurance in Ad hoc Networks 2003, pp. 384- 391, January 2003.

[151]. Girao, J., M. Schneider, and D. Westhoff. "CDA: Concealed Data Aggregation in Wireless Sensor Networks". in ACM Workshop on Wireless Security 2004, pp. 2004.

[152]. Yang, Y., X. Wang, S. Zhu, and G. Cao. "SDAP: A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks", ACM Transactions on Information and System Security (TISSEC), Vol. 11, Iss. 4,

[153]. "Homomorphic Encryption". Available at: http://en.wikipedia.org/wiki/Homomorphic_encryption, Access date:01/03/2007.

[154]. Peter, S., K. Piotrowski, and P. Langendoerfer. "On Concealed Data Aggregation for Wireless Sensor Networks". in Consumer Communications and Networking Conference (CCNC) 2007. Las Vegas Nevada, USA, pp. January 2007.

[155]. Schneier, B. "Secret and Lies, Digital Security in a Networked World": Wiley. 2000.

[156]. Yang, H., H.Y. Luo, F. Ye, S.W. Lu, and L. Zhang. "Security in Mobile Ad hoc Networks: Challenges and Solutions", IEEE Wireless Communications, Vol. 11, Iss. 1, pp. 38-47, 2004.

[157]. Hur, J., Y. Lee, H. Yoon, D. Choi, and S. Jin. "Trust Evaluation Model For Wireless Sensor Networks". in 7th International Conference on Advanced Communication Technology (ICACT) 2005, pp. 491-496, 2005.

[158]. Boukerche, A., X. Li, and K. EI-Khatib. "Trust-Based Security For Wireless Ad Hoc And Sensor Networks", Elsevier Computer Communications, Vol. 30, Iss. 11-12, pp. 2413-2427, 2007.

[159]. Boukerche, A. and X. Li. "An Agent-based Trust and Reputation Management Scheme for Wireless Sensor Networks". in IEEE Global Telecommunications Conference (GLOBECOM) 2005. St. Louis, USA, pp. 1857-1861, 2005.

[160]. Jung, I., B. Lee, N. Ha, K. Cho, Y. Choi, M. Choi, B. Lee, and K. Han. "An Energy Efficient Clustering Method for Wireless Sensor Networks". in 6th WSEAS International Confrence on Electronics, Hardware, Wireless and Optical Communications 2007. Corfu Island, Greece, pp. 139-144, February 2007.

[161]. Heinzelman, W.R., A.C. Chandrakasan, and H. Balakrishnan. "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", IEEE Transactions on Wireless Communications, Vol. 1, Iss. 4, pp. 660-670, October 2002.

[162]. Kifayat, K., M. Merabti, Q. Shi, and D. Llewellyn-Jones. "Topology creation and adaptive routing for Wireless Sensor Network in building environment". in 6th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet) 2005. Liverpool, UK, pp. June 2005.

[163]. Kifayat, K., M. Merabti, Q. Shi, and D. Llewellyn-Jones. "Dynamic Group-Based Key Establishment for Large-scale Wireless Sensor Networks". in First

International Conference on Communications and Networking in China (Chinacom) 2006. Beijing, China, pp. 25-27 October 2006

[164]. Agrawal, R., J. Kiernan, R. Srikant, and Y. Xu. "Order Preserving Encryption For Numeric Data". in ACM International Conference on Management of Data (SIGMOD) 2004, pp. 563 – 574, June 2004.

[165]. Acharya, M., J. Girao, and D. Westhoff. "Secure Comparison of Encrypted Data in Wireless Sensor Networks". in WiOpt 2005. Trentino, Italy, pp. 47- 53, April 2005.

[166]. Salhieh, A. and L. Schwiebert. "Evaluation of Cartesian-based Routing Metrics for Wireless Sensor Networks". in Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS) 2004, pp. 2004.

[167]. Kim, K., H. Kim, and K. Han. "A Zone-Based Method for Selecting Clusterheads in Wireless Sensor Networks". in RSCTC 2006, pp. 667-676,

[168]. Gupta, I., D. Riordan, and S. Sampalli. "Cluster-Head Election Using Fuzzy Logic For Wireless Sensor Networks". in IEEE Communication Networks and Services Research Conference 2005, pp. 255-260, 16-18 May 2005.

[169]. Islam, J., M. Islam, and N. Islam. "A-sLEACH: An Advanced Solar Aware Leach Protocol for Energy Efficient Routing in Wireless Sensor Networks". in IEEE ICN 2007, pp. 4-4, 22-28 April 2007.

[170]. Younis, M., K. Akkaya, and A. Kunjithapatham. "Optimization of Task Allocation in a Cluster–Based Sensor Network". in 8th IEEE Symposium on Computers and Communications (ISCC) 2003. Antalya, Turkey, pp. 329- 334, July 2003.

[171]. Chen, M. *Data Compression for Inference Tasks in Wireless Sensor Networks.* 2006, Binghamton University State University of New York.

[172]. Park, J., Z. Kim, and K. Kim. "State-Based Key Management Scheme for Wireless Sensor Networks". in WSNS 2005. Washington, DC, USA, pp. 819-825, 7th November 2005.

[173]. "Encyclopaedia terms". Available at: http://www.pcmag.com, Access date:13/10/2007.

[174]. Dam, K. W., and Herbert S. L., *Cryptography's Role in Securing the Information Society*, Washington, DC, National Academy Press, 1996.

[175]. Luo, J., P.Papadimitratos, and J.-P. Hubaux. "GossiCrypt: Wireless Sensor Network Data Confidentiality Against Parasitic Adversaries". in Sensor, Mesh and Ad Hoc Communications and Networks (SECON) 2008. San Francisco, CA, pp. 441-450, 16-20 June 2008.

[176]. Giani, A., T. Roosta, and S. Sastry. "Integrity checker for wireless sensor networks in health care applications". in Second International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) 2008, pp. 135 - 138, Jan. 30 2008-Feb. 1 2008.

[177]. Chiang, M.W., Z. Zilic, K. Radecka, and J.-S. Chenard. "Architectures of increased availability wireless sensor network nodes". in Proceedings. ITC 2004, pp. 1232 - 1241, 26-28 October 2004.

[178] Chang, C., J. Lee, and C. Chen. "On the Forward and Backward Secrecy of HLL Group Key Exchange Mechanism". in Proceedings of the Fifth International Conference on Computer and Information Technology. 2005. pp. 702 - 705, Washington, DC, USA.

[179] Sinha, A., A. Chandrakasan, "Dynamic Power Management in Wireless Sensor Networks", IEEE Design & Test, Vol. 18, Iss. 2, pp. 62-74, March 2001.