Energy Efficient Secure Routing in Wireless Sensor Networks

Kamanashis Biswas
MS(CS), BS(CSE)

School of Information and Communication Technology
Griffith Sciences
Griffith University

Submitted in fulfilment
of the requirements of the degree of
Doctor of Philosophy

March, 2016
“If a man will begin with certainties, he will end in doubts; but if he will be content to begin with doubts, he will end in certainties.”

Francis Bacon (1561-1626)
Declaration

This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Kamanashis Biswas
March 16, 2016
Abstract

Wireless Sensor Networks (WSNs) can contain thousands of small, inexpensive sensors that are randomly deployed in open and harsh environments to collect data. The short lifespan of the battery-operated sensors and the hostile environments necessitate the development of energy efficient and secure protocols in sensor networks. Among the wide variety of network protocols, routing plays the most significant role in energy consumption since 70% of the total energy is consumed for data transmission in WSNs. Therefore, it is necessary to design energy efficient routing schemes to conserve energy and prolong the network lifetime. However, resource limited sensors, lack of a global addressing scheme, and application-specific design of WSNs make routing a challenge. Furthermore, security is another critical issue in WSNs since sensors are generally deployed in unprotected environments and vulnerable to security attacks. The security algorithms have to be integrated with routing protocols to provide authenticity, confidentiality, and integrity of transmitted data. Most of the existing routing protocols implement different security mechanisms to achieve the security goals. Any conflict among these measures may create vulnerabilities in the network. Therefore, to ensure energy efficiency and minimisation of the implementation gap, energy efficient secure routing protocols have to be designed using a common security framework.

The harsh environment with severe resource constraints requires an energy efficient secure routing protocol to extend network lifetime and secure data communication. This thesis supports the claim conducting a detailed survey on existing literature. The first study develops a clique based clustering and routing (CBCR) protocol and an interference aware heuristic routing (IAHR) protocol for different network applications. The protocols improve network performance in terms of throughput, lifetime, and latency. Experimental results show that the CBCR protocol improves the network lifetime by an amount of 4% to 40% over the LEACH and EEAGA protocols. On the other hand, the IAHR protocol improves the network throughput by an amount of 10% over GAHR and 25% over the AODV Jr protocol. The second study develops a new key exchange mechanism and lightweight encryption algorithms that combine the benefits of the elliptic curve, chaotic map, and genetic operations to provide end-to-end data confidentiality. Performance evaluation on the Mica2 sensor mote shows that the CGEA cipher performs two to nine times better than the TWINE and LED encryption algorithms in terms of CPU
elapsed time. Finally, we develop an energy efficient secure multipath (EESM) routing protocol which integrates the security mechanisms with a core routing protocol. In order to design a common security framework, all security measures are implemented using the proposed cryptographic schemes which are tightly coupled with a proposed location-based routing protocol. Furthermore, EESM effectively defends against security attacks and improves the network lifetime by an amount of 25-37\% over the SEEM and SRMR protocols. In future work, we aim at implementing the proposed routing protocols in real sensors to evaluate their effectiveness and designing lightweight encryption algorithms to defend against advanced security attacks such as the quantum attack.
Acknowledgements

I would like to begin by thanking Associate Professor Vallipuram Muthukkumarasamy, my PhD supervisor and mentor for the past three and half years. He has been a great supervisor, providing me with support, guidance, and encouragement throughout my doctoral studies. I am grateful to Dr Muthukkumarasamy for the countless hours he spent with me, discussing everything from research problems to career goals, critiquing my ideas, reading my papers, and providing invaluable feedback on my research.

I would also like to thank Dr Xin-Wen Wu, my PhD associate supervisor. Dr Xin-Wen has been a wonderful supervisor. His breadth of knowledge, foresightedness, and enthusiasm for research has been an inspiration. His scholarly guidance, fruitful scientific discussions, and valuable feedback have greatly enhanced and strengthened my PhD research.

I would like to thank Dr Kalvinder Singh, my external supervisor, for his thorough feedback on my research work. His suggestions were invaluable and very helpful. I thank him for all the time and energy he has invested to provide his worthwhile technical feedback.

I am also grateful to Dr Elankayer Sithirasenan for his support and feedback during the first one and half year of my doctoral studies. I would like to thank Dr Elankayer for introducing me with WASP motes and encouraging me to conduct experiments on real sensors. While at Griffith University, I had the privilege of interacting and sharing my problems with a number of wonderful, bright, and talented people. I have learnt a lot from them. Their advice, critical feedback, and friendship have made my PhD experience more educational, more colourful, and more fun: Muhammad Usman, Alireza Jolfaei, Anisur Rahman, Zahra Jadidi, Bafrin Zarei, Sultan Alneyadi, Gul Zammen Khan, Patrick Crilly, Raihana Ferdous, and all members of the Network Security Group. It has been an immense pleasure to work with you. I would especially like to thank Usman and Gul for the brainstorming sessions on statistical analysis.

Finally, I would like to express my gratitude to my wonderful wife, Happy Mazumder. She has been my best friend, my “source of inspiration” since the day I met her almost twelve years ago. She keeps my life filled with joy and love, and I know that there is nothing I cannot accomplish with my wife by my side.
List of Publications

Journals


Conferences


# Contents

Declaration iv  
Abstract vi  
Acknowledgements viii  
List of Publications x  
List of Figures xvi  
List of Tables xx  

1 Introduction 1  
1.1 Wireless Sensor Network 1  
1.2 Motivation 5  
1.3 Objectives 7  
1.4 Thesis Outline 8  

2 Literature Review 9  
2.1 Introduction 9  
2.2 Routing Challenges and Design Issues 12  
2.3 Routing Protocols in WSNs 14  
2.3.1 Single Path Routing 15  
2.3.1.1 Flat-based routing protocols: 15  
2.3.1.2 Hierarchical routing protocols: 23  
2.3.1.3 Location-based routing protocols: 39  
2.3.2 Multipath Routing 45  
2.4 Security Issues in Routing Protocols 55  
2.4.1 Security Goals 55  
2.4.2 Attacks in WSNs 57  
2.4.3 Secure Measures for WSN Routing Protocols 59  
2.5 Research Challenges 69  

3 Methodology 73  
3.1 Research Questions 73  
3.2 Hypotheses 74  
3.3 Research Methodology 75  
3.3.1 Investigation Phase 76
5.3.6 Performance Comparison ................. 149
5.3.7 Discussion .................................. 150

5.4 Performance Evaluation on Hardware .......... 150
5.4.1 Implementation Environment ................. 151
    5.4.1.1 Hardware Specification ................. 152
    5.4.1.2 Software Specification ................. 152
5.4.2 Performance Evaluation and Analysis .......... 153
    5.4.2.1 Memory Consumption ................. 154
    5.4.2.2 Computational Cost ................. 154
    5.4.2.3 Operation time ..................... 155
5.4.3 Discussion .................................. 156

5.5 Summary ...................................... 157

6 Energy Efficient Secure Multipath Routing 159
6.1 Introduction .................................. 159
6.2 Network Model and Assumptions ................. 161
6.3 Energy Efficient Secure Multipath Protocol ........ 162
    6.3.1 Route Discovery: Pairwise Key Setup ........ 163
    6.3.2 Route Discovery: Neighbour Information Collection ........ 165
    6.3.3 Route Discovery: Coordinator Selection and Status Collection .... 166
    6.3.4 Route Discovery: Route Selection and Distribution .......... 168
    6.3.5 Data Transmission ..................... 169
    6.3.6 Network Maintenance ..................... 170
6.4 Lifetime Analysis of WSNs ..................... 172
    6.4.1 Modelling Lifetime ..................... 172
        6.4.1.1 Remaining Lifetime of an Individual Sensor ........ 173
        6.4.1.2 Remaining Lifetime of a Sensor Network ........ 176
    6.4.2 Lifetime Modelling: Single Path Protocol ........ 177
        6.4.2.1 Verification and Analysis of the Model .......... 178
        6.4.2.2 Simulation Results ................. 179
    6.4.3 Lifetime Modelling: Multipath Protocol ........ 182
6.5 Implementation of EESM Protocol ............... 184
    6.5.1 Security Measures ..................... 184
    6.5.2 Security Overhead Analysis ............... 185
    6.5.3 Network Setup Time ..................... 186
    6.5.4 Throughput and PDR Analysis ............... 188
6.6 Resilience to Security Attacks ............... 189
## List of Figures

1.1 Wireless sensor network architecture .................................. 2

2.1 MEMS markets by applications ........................................ 10
2.2 Perceived importance of WSN attributes ............................. 11
2.3 Relationship between different routing attributes .................. 12
2.4 Route construction in AODV ............................................ 16
2.5 Interest diffusion in sensor networks ................................... 17
2.6 Path creation by an agent ............................................... 18
2.7 Packets queuing mechanism .......................................... 22
2.8 Cluster formation process in LCA ................................... 24
2.9 Tree discovery in the HCC protocol .................................. 26
2.10 State transition for the FLOC clustering algorithm ............. 28
2.11 Dendrogram using the UPGMA with quantitative data .......... 34
2.12 Formatted clusters with cutting hierarchical tree .................. 34
2.13 Formatted clusters with minimum cluster size .................... 35
2.14 Cluster formation process in SDC .................................. 38
2.15 Greedy and perimeter routing in GPSR .............................. 39
2.16 Virtual grid in GAF .................................................. 40
2.17 Optimal routing path selection process in MERR ................. 42
2.18 Greedy forwarding in the GAHR protocol ......................... 43
2.19 Routing tables construction process in INSENS ................... 46
6.8 CBC-based message authentication code procedure .................................. 185
6.9 One-way hash chain generation procedure ................................................ 185
6.10 Network setup time for different network size ........................................... 187
6.11 Effects of node failures ........................................................................... 191
6.12 Effects of jamming attacks ...................................................................... 193
6.13 Effects of compromised node attacks ...................................................... 194
6.14 Compromised node detection .................................................................. 195
6.15 Comparison on network lifetime ............................................................... 197
6.16 Comparison on packet delivery ratio ......................................................... 198
6.17 Comparison on control overhead ............................................................... 199

A.1 The ATEMU interface ............................................................................... 215
A.2 A Mica2 Mote ......................................................................................... 216
A.3 An Arduino Pro Mote ............................................................................... 216
A.4 Point addition ........................................................................................... 217
A.5 Point doubling .......................................................................................... 218
A.6 The Pseudorandom bit sequences generation process ............................... 219
List of Tables

2.1 Comparison among SPIN, LEACH, and Directed Diffusion .................. 27
2.2 Functional comparison among HEED, LEACH, and LU-PEACH protocols ..................................................... 31
2.3 Functional comparison between PEGASIS and LA-PEACH protocols 31
2.4 Comparison of the leader-first hierarchical routing protocols ........... 36
2.5 Classification and comparison of the location-based routing protocols 45
2.6 Comparison among different lightweight cryptographic algorithms ... 65

4.1 Connectivity matrix of node 1 ................................................. 90
4.2 LEACH, CBCR, and EEGA simulation parameters ....................... 98
4.3 IAHR, GAHR, and AODV Jr. simulation parameters .................... 117
4.4 Comparison on routing overhead and latency ............................ 122

5.1 Memory occupation for RC5, Skipjack, and SLES .......................... 133
5.2 CPU consumption and elapsed time to encrypt 32 bytes data .......... 134
5.3 Correlation coefficients of adjacent pixels ................................. 146
5.4 NPCR and UACI test results .................................................. 147
5.5 Information entropy of plain and cipher images ........................... 148
5.6 Comparison on CPU and memory usage ................................... 149
5.7 Costs and hardware specifications (2014) ................................ 151
5.8 Cipher parameters used in experiments .................................... 152
5.9 Memory consumption of block ciphers in bytes .......................... 154
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.10 Computational cost of block ciphers in CPU cycles</td>
<td>155</td>
</tr>
<tr>
<td>6.1 Remaining lifetime of sensors at different locations</td>
<td>178</td>
</tr>
<tr>
<td>6.2 Remaining lifetime of the sensor network</td>
<td>179</td>
</tr>
<tr>
<td>6.3 Simulation parameters</td>
<td>180</td>
</tr>
<tr>
<td>6.4 Overhead of RC5 and CGEA</td>
<td>186</td>
</tr>
<tr>
<td>6.5 Throughput and PDR Analysis</td>
<td>189</td>
</tr>
<tr>
<td>6.6 EESM, SEEM, and SRMR simulation parameters</td>
<td>196</td>
</tr>
</tbody>
</table>
The application domains of WSNs have gradually expanded with the continuous development of wireless technologies, intelligent sensors, and micro-electronic-mechanical systems (MEMS). Sensors in real applications are often equipped with small memories, weak processors, limited battery power, and low transmission range. WSNs intended for various industrial, medical, and military applications, therefore, keep challenging researchers in the design of secure and energy efficient protocols. This chapter provides an overview of sensor networks, applications, and key challenges in developing communication protocols for WSNs. Motivation, objectives, and outline of the thesis are also presented.

1.1 Wireless Sensor Network

Wireless sensor networks consist of a number of sensor nodes (SNs), which monitor physical phenomenon such as temperature, pressure, humidity, motion, and sound [1]. These sensors are simple processing devices with limited computational capability, memory, and transmission range. A WSN can contain thousands of such small, inexpensive sensors, which are randomly deployed in open, unprotected and harsh environments for extended periods of time to collect data. These nodes either continuously or periodically send collected information to the base station (BS) via
an integrated radio transmitter. Given the vast area to be covered, the short lifespan of battery-operated sensors and the possibility of having damaged nodes during deployment, SNs are expected to be densely distributed in many WSN applications. They self-organise into a network through wireless communication and collaborate with each other to accomplish a common task. The capability of self-configuration and wireless communication allows them to be deployed in an ad-hoc fashion in remote, and hostile environments without any existing infrastructure. Even in a sparsely deployed sensor network, a SN can communicate with a far away node in the network through multi-hop communication. This characteristic of WSNs allows the expansion of the monitoring area through the addition of sensor nodes in the network. Figure 1.1 presents a WSN, where a SN is forwarding information to the BS via other SNs.

![Wireless Sensor Network](image)

**Figure 1.1: Wireless sensor network architecture**

Recent research forecasts that the global industrial WSN market will increase at a rate of 14.4% annual growth through 2016 [2]. The emerging wireless technologies enable the implementation of sensor networks for a wide range of applications such as for the military, healthcare, surveillances, environmental monitoring, structural health monitoring, automation, and industry control. Some of these applications are described below.
• **Military Applications:** Sensor networks have been successfully designed and built in military and national security applications over the past decades e.g., VigilNet [3]. WSNs are playing a vital role, being an integral part of military command, control, surveillance, reconnaissance, targeting systems, and unusual event detections. Distributed Sensor Networks (DSN) and Sensor Information Technology (SenIT) are two examples of successful implementation of WSNs in military sensing by the Defence Advanced Research Project Agency (DARPA) [4].

• **Environmental Monitoring:** Over the past few years, a number of environmental monitoring projects have emerged and attracted researchers worldwide. In this application, the sensor nodes monitor the environmental phenomena and relay their readings into a satellite link that allows the scientists to access real-time data over the Internet. Some examples are: Habitat Monitoring on Great Duck Island [5]; the Princeton Zebranet project [6]; the North Temperate Lakes project [7]; the Firebug project [8]; and vine monitoring in Pickberry Vineyard [9].

• **Medical Applications:** Many hospitals, clinics, and diagnostic centres are availing the benefits of WSN technologies to a range of medical applications e.g., Alarmnet [10]. There are many significant contributions of WSNs in healthcare systems. These include monitoring patient’s physiological data such as glucose level, blood pressure or heart rate, infant monitoring, elderly patient assistance, in-hospital emergency care, and disaster response. Furthermore, recent advances and developments of wearable sensors and biosensors are playing a pivotal role in the emergent body sensor network arena [11].

• **Structural Health Monitoring:** Structural health monitoring is a technology used to investigate the structural state of buildings and bridges [12] and detects structural change that may affect the performance of a structure. For example, a sensor network can help to monitor and measure the mechanical stress caused by natural disasters such as earthquakes or cyclones. UC-Berkeley engineers developed inexpensive tiny sensors called Smart Dust
motes which are capable of sensing a number of factors such as light, temperature, and dynamic responses [13].

- **Home Applications**: Home Automation (HA) is an example of home applications that facilitates flexible management of the home environment from anywhere at home or even from the outside, via the Internet. It increases the comfort level of residents and provides a distributed control of heating, ventilation, air conditioning, and lighting to save money and energy [14]. Recent research indicates that HA will be one of the fastest growth sectors in consumer electronics through 2020 [15].

- **Industrial Automation**: Industrial applications includes inventory tracking in stores or warehouses, automated machinery monitoring, fleet management, and factory process control. These applications provide conservation, control, efficiency, and safety through the optimisation of maintenance operations. As an example, Airbus’s A380 airplane uses passive RFID chips on removable parts (i.e., seatbelts, and plane components) to reduce inspection time [16].

In addition to high-end applications, current research interest has extended to ubiquitous high-reliability applications covering multi-threat management e.g., self-healing land mines [16]. Regardless of the disparity in the objectives of WSN applications, efficient and reliable protocols are required to accomplish the designated tasks. In practice, there are a number of challenges and hurdles in developing communication protocols for WSNs. They are as follows:

i) Limited functional capabilities

ii) Power factors

iii) Environmental factors

iv) Transmission channel factors

v) Topology management complexity

vi) Scalability concerns
The key challenge in WSNs is to maximise the lifetime of sensors since it is not feasible to replace or recharge the batteries of thousands of SNs deployed in a remote area. Therefore, energy efficient mechanisms must be adopted in both computational operations of SNs and communication protocols. Data aggregation and routing are two major techniques that can remarkably reduce energy consumption in WSNs by eliminating data redundancy and communication overheads. Since data transmission contributes 70% of the total energy dissipation in a sensor network, routing protocols play a significant role in energy consumption [17]. This is one of the reasons behind extensive research and development of energy efficient routing protocols in WSNs.

Data communication security is another major concern in many WSN applications. Specifically, the uses of sensors in critical systems such as nuclear power plants, aircrafts, battle-fields, and hospitals require effective mechanisms to ensure the authenticity, confidentiality, and integrity of transmitted data. Thus, in order to provide secure communications in WSNs, it is crucial to design and develop lightweight protocols by proper utilisation of available resources. Furthermore, security protocols should work together with routing protocols to avoid any conflict between them. A conflict in implementing security and routing algorithms may lead to vulnerabilities in network security.

1.2 Motivation

There has been an increased interest in the potential collaboration among sensors in data aggregation and processing, coordination, and management of sensing activities in recent years. However, the resource constrained nature of sensor nodes poses a number of barriers in designing, operating, and maintaining sensor networks in the real environment as well as necessitates energy awareness at all layers of the networking protocol stack. For example, energy efficient mechanisms at network layer are highly desirable for route discovery and relaying of data to prolong the network lifetime, whereas lightweight cryptographic schemes are preferred to conserve node energy at the application layer.
There are a number of factors that make routing more challenging in WSNs compared to mobile ad-hoc networks or cellular networks. First, it is not possible to build a global addressing scheme (e.g., IP-addressing) in sensor networks due to the maintenance overheads caused by a large number of sensor nodes. Second, sensor networks are data centric and application specific. Therefore, the design requirements of sensor networks change according to their applications. For example, the requirements of time-critical applications for early warning are different from that of long-term data collection in an environmental monitoring system. Third, due to topology changes, interference may be caused by environmental influences or adversaries. Therefore, the routing scheme has to be fault-tolerant and adaptive so that a complete breakdown of the network can be avoided by using up-to-date routing information.

With the increasing number and variety of mission-critical applications, security has become a major concern in WSNs. Since sensors are usually deployed in unattended environments, they are vulnerable to a number of security attacks. Furthermore, there are a number of challenges that make secure communication hard to achieve in WSNs. First, due to the broadcast nature of the wireless medium, an adversary can easily eavesdrop, access or tamper transmitted messages illegally. Moreover, the attacker may launch a denial-of-service (DoS) attack or rushing attack using the captured broadcast messages [18]. Second, it is not possible to implement complex and computation-intensive cryptographic schemes on SNs because of their limited resources. This limitation keeps the door open to external attacks by powerful adversaries having more computing and communication equipment. Third, WSNs are highly susceptible to physical capture. If a node is compromised, the attacker can extract all secret information and use them to launch insider attacks. Hence, it is important to detect compromised nodes immediately in order to limit, or localise, the damage caused by the attacker. Fourth, to protect WSNs from adversaries, several security measures such as One-way Hash Chain (OHC), Message Authentication Code (MAC), key exchange, and encryption schemes are combined together which may use completely different parameters, design structures, and implementation strategies. However, the use of different security mechanisms may lead
to the implementation gap, which can be exploited by adversaries. Fifth, implementation of security mechanisms can introduce further delay and communication cost. Depending on the required level of security, there must be a trade-off between cost and security. Therefore, to provide a sufficient level of security, it is important to design efficient and lightweight security protocols in WSNs.

1.3 Objectives

This thesis aims to provide effective solutions for two fundamental problems of WSNs: i) design and development of energy efficient routing protocols for small and large-scale WSNs, and ii) design and integration of lightweight security algorithms with routing protocols to ensure secure communication in the network. Precisely, the objectives of this dissertation are:

i) designing energy efficient single path and multipath routing protocols to extend the network lifetime. The clique-based hierarchical routing approach and the interference aware heuristic approach are implemented to develop new routing schemes.

ii) developing a new key exchange mechanism and lightweight encryption algorithm to provide secure communication. The security features of genetic operations, chaotic map, and elliptic curve are exploited to design new efficient cryptographic schemes for WSNs.

iii) minimising the implementation gap of different security measures by proposing a common security framework. A number of security measures are implemented using the same underlying structure to reduce security vulnerabilities in the network.

iv) designing analytical models for lifetime estimation of wireless sensor networks. In addition to experimental analysis, an analytical model is proposed that can be used to predict the lifetime for both single path and multipath routing protocols.
1.4 Thesis Outline

Chapters in this dissertation are organised as follows:

Chapter 1 provides a brief introduction, and outlines the motivation and objectives of this thesis.

Chapter 2 presents the relevant background and related literature of the research topic. The advantages and limitations of both single path and multipath routing protocols are described. Furthermore, the limitations of conventional and lightweight encryption algorithms are explained in this chapter.

Chapter 3 identifies the research questions on the basis of limitations described in Chapter 2. Furthermore, the chapter also defines the hypothesis and describes the methodologies used to validate the hypothesis.

Chapter 4 describes single path routing protocols that use multi-hop communication to route data packets. An energy efficient clique-based hierarchical protocol for large-scale sensor networks, as well as an interference aware heuristic routing protocol for small-scale networks, are presented in this chapter.

Chapter 5 presents node verification and encryption schemes for resource constrained sensor nodes. A simple node verification method, based on elliptic curve points, is proposed. Furthermore, two lightweight encryption schemes are designed to provide an enhanced level of security with minimum resources.

Chapter 6 proposes a new multipath routing scheme that combines security measures and routing protocols to provide secure communication in the network. An analytical model is designed to estimate the network lifetime and a real implementation is conducted to examine the suitability of different security measures.

Chapter 7 summarises the major contributions of this research, and outlines the future directions.
This chapter presents a critical review of relevant research reported in the literature. The pros and cons of the existing protocols are described in detail. Section 2.2 of this chapter briefly describes the routing challenges and design issues. Section 2.3 presents the classification of routing protocols and also investigates the benefits and limitations of single path and multipath routing protocols. Section 2.4 focuses on security issues in routing protocols. Section 2.5 details the research challenges identified in this thesis.

2.1 Introduction

Low cost wireless technology is opening up new possibilities for sensor networks and the application area of WSNs is rapidly growing over time. Today sensors are everywhere. They are in our mobile phones, vehicles, smart homes, and even in wearable devices. Although research on WSNs started back in 1980s, it has been received an increased interest from industry and academia since 2001. Recent research forecasts that WSN market is expected to grow more than $2 billions in 2021. This is due to the availability of small, low cost, and low power sensors. Figure 2.1 forecasts the growth of MEMS markets by applications for next five years [19]. It can be seen that consumer electronics is the main driving force since it contributes in more than 60% increase of total MEMS markets.
Chapter 2. Literature Review

Figure 2.1: MEMS markets by applications

With the advancement of wireless technologies, sensor networks become more ubiquitous, and more challenges are posed in the design and implementation of WSNs. As illustrated in Figure 2.2, reliability, lifetime, and cost are three primary design considerations for many WSN applications such as industry controls [20]. Among these attributes, reliability and lifetime matter most to the customers. Without providing solutions for lifetime and reliability issues, cost is not yet a customer priority. Thus, one of the primary design goals of many WSN applications is maintaining a good trade-off between these two attributes. The work in this thesis also investigates the impacts of energy consumption and security issues that play a significant role in the design of routing protocols.
Since data transmission consumes more energy than any other activities in WSNs, it is important to design energy efficient routing protocols to prolong the network lifetime. Furthermore, it is required to preserve the integrity and confidentiality of transmitted data in many WSN applications. Therefore, energy and security are two critical concerns that have significant impacts on routing protocols.

Figure 2.3 illustrates the relationship between these two parameters, and how they affect computation and communication costs. Energy in WSNs is mainly consumed for computation and communication purposes. Computational energy cost deals with the energy use for data aggregation and execution of algorithms, whereas communication cost is the energy cost of transmitting and receiving information. Since transmitting one bit of information in WSNs dissipates about as much power as executing 800-1000 instructions, routing protocols should keep communication overheads as low as possible [21]. However, the implementation of security measures incurs cost in the routing protocol design. The use of complex cryptographic schemes will increase the computational cost, whereas communication cost will also be increased due to the exchange of security parameters and additional control messages. Therefore, routing protocol design should maintain a trade-off between cost and performance in order to achieve the security goals as well as energy efficiency.
2.2 Routing Challenges and Design Issues

Routing in sensor networks is more challenging than the conventional routing in fixed networks for a number of reasons. First, there is no infrastructure and thus the SNs should have self-organisation capabilities to keep the network functional. Second, failure of SNs is very common in WSNs due to unreliable wireless links. Therefore, the network must be reliable and fault-tolerant. Third, routing protocols have to be energy efficient in order to extend the lifetime since SNs are resource limited. Finally, SNs are deployed in hostile environments, and thus they are vulnerable to security threats. Therefore, routing protocols should be robust and secure to effectively defend against the attacks. The challenges and design issues having significant impacts on performance of routing protocols are summarised below [22].

Energy Consumption: As most of the sensor nodes are battery-driven, power conservation and power management are two important issues in WSNs. Energy aware protocols and algorithms for sensor networks are required to prolong the battery life.

Reliability and Security Considerations: Reliable data transmission and data
protection is a major challenge in sensor networks. Since sensor nodes may be deployed in a harsh environment, reliable and secure communication protocols must be designed to protect an adversary. Error-correction coding, retransmission mechanisms, encryption, and authentication schemes can be implemented to ensure data security but they also impose major challenges in terms of the cost of power and bandwidth.

**Network Dynamics:** In general, it is assumed that sensor nodes and the base stations are stationary in WSN applications. Therefore, very few protocols consider mobility as an important property of a sensor network. Hence, it is a challenge to implement those static protocols in dynamic environments.

**Node Deployment:** Nodes in WSNs need to apply self-organising and self-healing intelligence to continuously adapt to unpredictable environments if they are deployed randomly in a geographic location. In such situation, the network formation process requires effective routing mechanisms to keep energy dissipation minimum.

**Data Aggregation:** Data aggregation is a process of eliminating redundant data. To conserve energy, the gateway nodes perform the data aggregation process that helps to reduce network traffic. However, the use of secure data aggregation process is highly recommended to ensure data confidentiality and integrity in a critical system.

**Hostile Environment:** Sensor nodes can be deployed in an open, unprotected and hostile environment such as battlefields. In this situation, an adversary could compromise a sensor node or even could introduce his own malicious nodes inside the network. The adversary can capture sensitive information and use them for illegal purposes. Hence, designing resilient security protocols is a key challenge for researchers to protect sensitive information from compromised nodes.

**Random Topology:** In many applications, deploying a sensor network in a specific area is done by random distribution of sensor nodes e.g., from an airplane. In this situation, the nodes must be capable of self-organising themselves to build the communication topology. Since the neighbourhood cannot be known a priori, key
agreement protocols are required to prevent malicious nodes from being members of the network.

**Fault Tolerance:** A sensor node may fail due to physical damage, technical problem, or out of energy. If some nodes fail, the rest of the network must continue its operation without a problem. Adaptable protocols are required to overcome the problems of node failures or link congestions. As an example, the routing protocols must find alternative paths in the case of failure of a forwarding node.

**Scalability:** Many WSN applications are needed to deploy thousands of sensor nodes to collect data from the environment. Hence, the communication protocols must be designed in such a way that addition of more nodes in the existing network does not affect the clustering, routing, and network operations.

**Communication Latency:** Multi-hop routing, network congestion, and data aggregation in intermediate nodes may lead to a high latency in wireless sensor networks during the communication process. This makes synchronisation very difficult as some of the security mechanisms like cryptographic key distribution depend on the synchronisation of the participating nodes.

### 2.3 Routing Protocols in WSNs

Based on the communication strategy, routing protocols can be classified into two categories: *single path routing* and *multipath routing*. In single path routing, only a single path exists between the source node and the destination node. Single path routing protocol is simple since the route between two nodes can be constructed in a specific period of time. Furthermore, it is scalable, because the addition of large number of SNs does not affect the route discovery procedure. On the other hand, multipath routing constructs more than one path to deliver data from source to destination. Since multipath routing uses redundant paths in data transmission, it can largely address the load balancing, security, and reliability issues. However, single
path routing is suitable for many WSN applications due to its simplicity, scalability, and low communication overheads, whereas multipath routing is preferable for applications that require reliability and security. Furthermore, these protocols can be divided into proactive, reactive, and hybrid protocols depending on their route discovery techniques. The proactive protocols compute all routes before they are needed, whereas reactive protocols execute routing on demand [23]. Hybrid protocols are a combination of these two mechanisms. The following two sub-sections investigate the characteristics and limitations of single path and multipath routing protocols.

2.3.1 Single Path Routing

Depending on the network structure, routing protocols are divided into three categories: i) flat-based, ii) hierarchical, and iii) location-based protocols. In flat-based routing, nodes are typically assigned same responsibilities, whereas nodes perform different roles in the network in hierarchical routing. Unlike these two protocols, location-based routing uses sensor nodes’ position to forward data in the network. This section describes flat-based, hierarchical, and location-based single path routing protocols in details.

2.3.1.1 Flat-based routing protocols:

Flat-based routing protocols distribute information as needed to any node within communication range. They enable the delivery of packets among sensor nodes through any available path without considering network architecture. The following summarises the advantages and performance issues of flat-based single path routing protocols.

Ad-hoc On-Demand Distance Vector (AODV): AODV is a widely used routing protocol proposed for mobile ad-hoc networks [24]. It has also been adopted to other wireless networks for its simplicity and scalability. As an example, Zigbee networks implement the simplified version of the AODV protocol known as AODV
junior (AODV$_{jr}$) [25] which enables multihop routing to deliver data packets. When a node needs to send data packets, it broadcasts a Route Request (RREQ) message including source ID, destination ID, sequence number, and time-stamp. Upon receiving the message, a neighbour node checks whether it has a route to reach the destination. If the node finds a route to the destination, it sends a Route Reply (RREP) message back to the source node. Otherwise, it rebroadcasts the RREQ message. Figure 2.4 shows the route construction phase of the AODV routing protocol.

The AODV routing protocol has a number of limitations. First, it consumes a significant amount of energy because of flooding route request messages. Second, it does not reuse the routing information, and thus AODV results in a high traffic in the network. Finally, it incurs message overheads since the reverse path contains the addresses of all intermediate nodes of that path. However, researchers proposed an improved version of the AODV protocol to overcome some of these limitations.

Sensor Protocols for Information via Negotiation (SPIN): SPIN is a family of adaptive flat-based routing protocols which use a data negotiation and resource-adaptive algorithm [26, 27]. This family of protocols disseminate information to every node in the network as they consider all nodes as the potential base stations. A user can request information from any node in the network and receive the required information since all nodes in the network possess the same information.
One of the advantages of SPIN protocol is that topological changes are localised since each node requires to know only its single-hop neighbours. Unlike conventional flooding or gossiping based routing protocols [28, 29], SPIN is designed to save energy through data negotiation and elimination of redundant data. However, the data advertisement mechanism in SPIN cannot guarantee the delivery of data.

**Directed Diffusion:** Directed diffusion is a popular data aggregation paradigm which supports both single path and multipath data delivery [30]. It is a data-centric and application specific paradigm that combines data coming from different sources by eliminating redundancy, minimising the number of transmissions, and prolonging the network lifetime. In directed diffusion, the BS requests data through broadcasting interests where interests specify the task need to be carried out by the network. Then, interests are forwarded through the network hop-by-hop and gradients of information are setup to draw data satisfying the query towards the requesting nodes. Figure 2.5 shows an example of the working procedure of directed diffusion i.e., sending interests, building gradients, and data dissemination.

![Directed Diffusion Diagram](image_url)  
**Figure 2.5:** Interest diffusion in sensor networks
Directed diffusion differs from the SPIN protocol as it issues only on demand data queries. All communication in directed diffusion is neighbour-to-neighbour, where each node has data aggregation and buffering capability. Unlike SPIN, there is no need to maintain a global network topology in directed diffusion. However, directed diffusion is not suitable for applications which require continuous data delivery to the BS e.g., environmental monitoring. This is because the query-driven on demand data model may not applicable in this situation.

**Rumor Routing:** Rumor routing is a variation of directed diffusion and can be applied for applications, where the geographic routing is not suitable [31]. Instead of flooding the entire network with queries, the idea is to route only to the nodes that have observed a particular event to retrieve information about the occurring events. The protocol employs long-lived packets, called agents, in order to flood events through the network as shown in Figure 2.6. When a node detects an event,
it includes the event to its events table and generates an agent. These agents propagate information about local events to the distant nodes in the network. If a node generates a request for an event, the other nodes which know the route may respond to the request by inspecting their events tables. In this way, rumor routing eliminates the need of flooding the entire network and reduces communication cost.

Simulation results show that rumor routing improves energy saving compared to flood-based routing and can also handle node failures. But, it is not feasible for a large number of events as the cost of maintaining agents and event-tables in each node is very high.

**Random Walks:** Routing protocols with random walks is designed to achieve load balancing in large scale sensor networks [32]. To find a route from a source node to a destination node, the location information is obtained by computing the distances between two nodes using the Bellman-Ford algorithm. According to a computed probability, an intermediate node is selected as the next hop if it is closer to the destination among all other neighbour nodes. The main advantage of this protocol is that it is very simple and maintains a little state information. Moreover, it provides load balancing since different routes are chosen at different times even for the same source and destination nodes. However, the main concern about this protocol is that the topology of the network may not be practical.

**Simple Energy Efficient Routing:** The Simple Energy Efficient Routing (SEER) protocol [33] achieves energy efficiency by using the hop-count, residual energy of SNs, and distance between a SN and the BS. This flat-based routing algorithm is initiated by the source node and uses a uniform network to achieve energy efficiency. According to the authors, a good level of energy efficiency is achieved if the sensors are uniformly distributed and the BS is placed at the centre of the network. This indicates that the protocol is not suitable for randomly deployed WSNs.

**Energy Efficient Dynamic Source Routing (E2DSR):** E2DSR is based on the Dynamic Source Routing protocol and uses a new structure for control packets [34]. Moreover, the protocol changes routing behaviours of the sensor nodes, implements
a new ‘Energy Table’, and creates a new algorithm to maintain cached routes and selection. The route selection process of E2DSR employs a Route Priority Function that has three input parameters: i) length, ii) freshness, and iii) energy of the path. First, it determines the length of a path using the following equation:

$$L(i) = \frac{\text{Length of Route}(i)}{\text{Max Length}}$$

(2.1)

where \(\text{Length of Route}\) denotes the length of \(i\) in number of hops, and \(\text{Max Length}\) is the maximum length of a route in the DSR algorithm. The default value of \(\text{Max Length}\) is 16. Then, the protocol determines the freshness value of a route. The freshness parameter \(F(i)\) indicates how fresh a route is. The following equation is used to measure the freshness of route \(i\):

$$F(i) = \frac{n - i + 1}{n}$$

(2.2)

If there are \(n\) routes then the most recently used route will have a freshness value of 1, whereas the freshness value of the oldest route will be \(1/n\). For all other routes, the value is between this two limits. The final task is to determine the energy level of a specific route. The E2DSR protocol defines the following equation to measure the energy level \(E(i)\) of a route \(i\):

$$E(i) = \frac{\text{RE}(i) - \text{MRE}(i)) \cdot \text{MRE}(i)}{M(i) \cdot \text{InitialEnergy}^2}$$

(2.3)

where \(M(i)\) is the total number of nodes in route \(i\), InitialEnergy is the energy when a node is deployed, \(\text{RE}(i)\) is the remaining energy of route \(i\), and \(\text{MRE}(i)\) is the minimum remaining energy between all nodes of route \(i\).

Now, combining the three parameters, the protocol determines a path priority value for each route in order to select the most efficient route. The priority of a route \(i\) can be determined as follows:

$$\text{PathPriority}(i) = \frac{K_F \cdot F(i) + K_E \cdot E(i)}{K_L \cdot L(i)}$$

(2.4)
where $K_L$, $K_E$, and $K_F$ are the coefficients for length, energy, and freshness of the route respectively.

E2DSR employs a simple on demand route discovery mechanism that uses small routing tables to compute the priority factor of the routes. However, although the protocol is capable of balancing power consumption amongst different nodes in the network, it incurs both storage and communication overheads.

**Energy-efficient Asynchronous low Duty-Cycle Routing (E-ADCR):** E-ADCR is an efficient flooding-based routing protocol, which is combined with an asynchronous, blind, and opportunistic MAC protocol operating at a low duty-cycle [35]. In E-ADCR, the BS first broadcasts a beacon including a hop count parameter, i.e., $hop = 0$ initially. When a node hears this beacon with higher reception power than the given robustness threshold, it records the $hop$ value and rebroadcasts it with the incremented $hop$ value. Thus, on completion of the initialisation phase, every node has an estimation of its number of hops to the BS. This hop value is used to set the TTL (Time To Live) parameter for packets.

The E-ADCR protocol maintains a queue as shown in Figure 2.7. Every packet has a limited queue lifetime ($maxQueueTime$) and it will be removed from the queue if the lifetime expires. When a node receives a packet, it processes the packet if the node is the destination node. Otherwise, it checks whether the packet is new one and the TTL value is greater than 0. If both conditions are fulfilled, it records the packet in the queue and decrements the TTL value. If a node is active and requires to send packets, it removes the packets from top to bottom of the queue and broadcasts them one by one. This process is continued until the node goes to sleep mode. For example, in Figure 2.7(a), the node will broadcast packets in a repeated way ($p_3, p_2, p_1, p_3, p_2, \ldots$) as long as the node is active.

Although the E-ADCR protocol takes advantage of simplicity and efficiency of the shortest path, it is not suitable for mission critical applications in WSNs. Furthermore, the flooding-based mechanism consumes a significant amount of node energy, and thus leads to quick network partitions.
Improved Ad-hoc On-Demand Distance Vector (IAODV): The IAODV protocol, based on the existing location-aided routing (LAR) protocol, is proposed to reduce the number of RREQ packets [36]. The protocol bounds the search area by implementing the LAR control routing lookup strategy. Furthermore, the algorithm uses a path selection function to optimise power consumption and extend the network lifetime. The following definitions are used in IAODV to define the path selection function.

Definition 1: A route is expressed as $r_i = a_1, a_2, a_3, \ldots, a_j$, where $a_1$ is the source node, $a_j$ is the destination node, and $N$ is the number of hops from the source node to the destination node.

Definition 2: $E_i$ is the residual energy of $a_i$, the average residual energy of each node on the route $r_i$ is:

$$f(r_i) = \frac{\sum_{i=1}^{i} E_i}{N + 1}$$

(2.5)

Definition 3: The path selection function can be defined as:

$$f(r_{max}) = \max \{f(r_i) | r_i \in R\}$$

(2.6)

where $R$ is the set of all paths between source and destination nodes. The aim of
the function is to select a path that has less hops and more residual energy. For more than one path having the same residual energy, the path with the least hop is selected, whereas the path with the highest residual energy is selected if many paths have the same hop count.

Remarks: The advantages of flat-based single path routing protocols are that they are simple, scalable, structure-independent, storage efficient, and suitable for large networks. However, they create delays and consume a significant amount of energy in forwarding data packets because of the flood-based route discovery mechanism. Thus flat-based single path routing is less energy efficient compared to hierarchical and location-based routing protocols.

2.3.1.2 Hierarchical routing protocols:

Hierarchical or cluster-based routing is an efficient way to reduce energy consumption within a cluster. Unlike flat-based routing, cluster-based approach divides the network into different groups and assigns functionalities among the members of each group. Hierarchical routing basically works in two phases. In the first phase cluster-heads (CHs) are selected using some mechanisms while the second phase deals with route discovery techniques. Based on the order of the cluster formation and the CH selection processes, the clustering protocols can be divided into two categories: the leader-first approach and the cluster-first approach.

Leader-first Approach: In the leader-first approach, cluster heads are selected first based on some attribute metrics (e.g., best energy level, maximum connectivity, data attributes) and then they collaborate on how to assign other nodes to different clusters. The following are some examples of leader-first protocols.

Linked Cluster Algorithm (LCA): LCA is one of the early routing algorithm in wireless networks [37–39]. This clustering approach is divided into five phases. The first phase is topology sensing, where each node discovers its neighbours using a probe message. A probe message is a broadcast message sent by each node. When the other nodes hear the message, they send an acknowledgement back to
the probing node. By this query method, each node discovers all nodes to which it is bidirectionally connected. The next two steps of LCA are cluster formation and cluster linkage. In this phase, cluster heads as well as cluster members are identified and then communication links are established among different clusters. The way of selecting a cluster head is very simple. At first, a node is selected arbitrarily as a cluster head. A circle is drawn around the selected nodes, where the radius is equal to the communication range of the selected node. All nodes within this range become members of the selected node, and thus they form a cluster as shown in Figure 2.8(a).

After formation of the first cluster, another node is selected randomly, and the previous procedure is followed again. The only constraint is that if a node is already a member of a cluster then it cannot be a member of another new cluster. After the selection process of cluster heads and members, the gateway nodes are selected to perform the inter-cluster communication. The overlapping nodes get priority to act as gateway nodes when two clusters are overlapped with each other as shown in Figure 2.8(b). On the other hand, in the case of non-overlapping adjacent nodes, one node from each cluster is selected using some deterministic rules such as the lowest sum of identity numbers as depicted in Figure 2.8(c). The final two steps are link activation and routing. The link activation algorithm applies the Frequency
Hopping Code Division Multiple Access (FH-CDMA) mechanism with the Time Division Multiple Access (TDMA) to establish a single code channel and channel access control.

The benefits of the LCA protocol is that it is fault-tolerant and provides maximum network connectivity and node mobility. However, the protocol is not energy efficient and low-energy nodes may be selected as CHs.

**Threshold sensitive Energy Efficient Sensor Network protocol (TEEN):**
TEEN is the first protocol developed for reactive networks, which is used in temperature sensing applications [40]. It has two additional restrictions. First, when the absolute value of the sensed attribute is beyond a Hard Threshold ($H_T$), the node sensing this value must switch on its transmitter and report it. Second, when a change in the value of the sensed attribute is larger than a Soft Threshold ($S_T$), it triggers the node to switch on its transmitter and report the sensed data. A node will report data only when the sensed value exceeds $H_T$, or the change in the value is larger than $S_T$. The main drawbacks of this scheme are: i) it is not suitable for real time applications; and ii) the practical implementation needs to ensure that there are no collisions in the cluster. The TDMA scheduling can be used to avoid this problem, although this will cause delays in the reporting of time-critical data [41].

**Hierarchical Control Cluster (HCC):** Hierarchical control clustering algorithm is a multi-tier clustering scheme. The cluster formation process of HCC protocol consists of two phases: tree discovery and cluster formation [42]. At the beginning of the algorithm, the root node is selected. Any node in the WSN can initiate the process and claim to be the root node. If multiple nodes initiate the process then the least node ID will be selected as the root node. After that, the tree discovery process starts. Each node $u$ transmits a tree discovery beacon that contains the following fields: src-ID, parent-ID, root-ID, root-seq-no, and root-distance. Receiving the beacon, if any neighbour node $v$ of $u$ discovers a shorter path to the root through $u$, then it will update its path as shown in Figure 2.9. Every node updates its sub-tree size when its children sub-tree size changes. The complete tree will be formed in this way. The cluster formation phase starts when a sub-tree on a node crosses the
size parameter, $k$. The node initiates the cluster formation process on its sub-tree. It will form a single cluster for the entire sub-tree if sub tree size is less than $2k$, or else, it will form multiple clusters.

**Figure 2.9:** Tree discovery phase of the cluster creation

HCC protocol considers the cluster size and the degree of overlapping as important attributes in the cluster formation process. Thus, it balances the network traffic effectively and is suitable for dynamic environments. It also provides node mobility, stability, and recovering strategies in case of node failures. Still, it has some pitfalls. First, the protocol is complex in architecture. Second, if a change occurs at the downwards level due to node mobility or node failures, then huge changes have to be made in order to rebuild the BFS (Breadth First Search) tree. Third, the approach is not energy efficient.

**Low Energy Adaptive Clustering Hierarchy (LEACH):** Low Energy Adaptive Clustering Hierarchy (LEACH) is a clustering-based protocol which forms clusters on the basis of the received signal strength and uses the cluster head nodes as routers to communicate with the base stations [43]. During the setup phase, a predetermined number of nodes, $p$, announce themselves as CHs. A sensor node picks a random number, $r$, between the range 0 and 1. If this number is less than a threshold
value $T(i)$, the node becomes the CH for the current round. The threshold value is measured based on an equation that incorporates the fraction of nodes willing to be CHs, the current round, and the number of nodes that have not been selected as a CH in the last $(1/p)$ rounds, denoted by $G$. The equation can be expressed in the following form:

$$T(i) = \frac{p}{1 - p(r \mod (1/p))} \text{ if } i \in G$$

where $G$ is the set of nodes that are involved in the CH election set. The elected CHs broadcast their status of being selected as CHs to all other nodes in the network. Now, each non-CH node finds out its cluster by selecting the CH that can be reached by using the lowest communication energy. LEACH also performs periodical rotation of cluster heads in order to balance the load. The main limitation of LEACH is that the selection of cluster head is probabilistic. Therefore, it is more likely that a node with low energy may be selected as a CH. When this node runs out of the power, the whole cluster becomes dysfunctional.

The following table shows the comparison among the SPIN, LEACH and the Directed Diffusion routing protocols for different parameters [23]. Due to in-network processing, the performance of the Directed Diffusion protocol is better than the other two protocols in terms of energy efficiency.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>SPIN</th>
<th>LEACH</th>
<th>Directed Diffusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal Route</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Network Lifetime</td>
<td>Good</td>
<td>Very Good</td>
<td>Good</td>
</tr>
<tr>
<td>Resource Awareness</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Use of Meta-Data</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Power-Efficient Gathering in Sensor Information Systems (PEGASIS):**

PEGASIS is a chain based protocol which reduces energy dissipation and elongates the network lifetime by making nodes to communicate only with nearest neighbours [44]. It is assumed that every node should know the location information of all other nodes. PEGASIS starts with the furthermore node from the BS. The chain
can be constructed easily by using a greedy algorithm. The chain leader accumulates data and transmits it to the BS. Each node in the chain takes turn to be the leader in order to balance the overhead involved in communication between the BS and the chain leader. The drawback of PEGASIS is that the protocol makes certain assumptions that are hard to ensure. Firstly, each node is capable of communicating with the BS, whereas SNs generally use the multi-hop routing. Secondly, all nodes have the same level of energy, therefore, they exhaust at the same time. Finally, excessive data delay is another limitation of PEGASIS. However, the modified version of the PEGASIS protocol improves the energy efficiency by using the concentric clustering mechanism [45].

**Fast Local Clustering Service (FLOC):** In FLOC, the communication range is divided into two categories: *i-band* (inner band) and *o-band* (outer band) as shown in Figure 2.10 [46]. It assumes that nodes within *i-band* range (signal power \([0.5, 1]\)) will receive all messages sent among them while some of the messages may be lost for the nodes of *o-band* range (signal power \([0.2, 0.5]\) ). In this method, a node stays idle for a random period of time to receive an invitation from any potential CH. If it does not get any request, it becomes a candidate CH and broadcasts a candidacy.

![State transition for the FLOC clustering algorithm](image)
message. If any other node (suppose $k$) receives the candidacy message, which is already a member of a cluster, will reply back to the candidate CH about its cluster information ($C_k$). Then, the candidate CH will join $C_k$ as an o-band node. If the candidate node does not receive any conflict message, it becomes a cluster head and invites other nodes to join in its cluster. Any o-band node that receives an invitation from a closer CH later, can switch its membership to that new cluster.

FLOC forms relatively equal sized clusters with minimum overlapping and provides scalability and self-healing capabilities. Furthermore, the overhead of the cluster formation process is very low in FLOC. The pitfall of the approach is that energy efficiency and data dissemination is not described clearly.

**Hybrid Energy-efficient Distributed (HEED) Algorithm:** HEED is a distributed algorithm which considers energy and communication cost when selecting CHs [47]. The algorithm is divided into three phases:

1. **Initialisation Phase**– To limit the initial CH announcement, the algorithm sets an initial percentage of CHs ($C_{prob}$) among all sensors at first. Each node determines its probability of becoming a CH as follows:

$$CH_{prob} = \frac{C_{prob} \times E_{residual}}{E_{max}} \quad (2.8)$$

where, $E_{residual}$ is the current energy of a SN, $E_{max}$ is the maximum energy, which corresponds to a fully charged battery. $CH_{prob}$ is not allowed to fall below a certain threshold $p_{min}$, which is selected to be inversely proportional to $E_{max}$.

2. **Repetition Phase**– In this phase, CHs with least transmission cost are selected through several iterations. If a node does not hear from any CH, the sensor elects itself as a CH and sends an announcement message to its neighbours. Finally, each sensor doubles its $CH_{prob}$ value and goes to the next iteration of this phase. It stops when the $CH_{prob}$ value reaches to 1. Therefore, two types of cell-head status that a node could announce to its neighbours are:

- **Tentative status:** The node becomes a tentative CH if its $CH_{prob}$ less than 1. It can change its status to a regular node at a later iteration if it finds a lower cost
30

Chapter 2. *Literature Review*

Final status: The node becomes a CH if its $CH_{prob}$ has reached to 1.

3. Finalisation Phase—In this phase, each sensor makes a final decision whether it will select the least cost CH or it will announce itself as a CH.

HEED is an energy efficient and balanced clustering scheme. Moreover, it provides scalability and stability. One disadvantage of the protocol is that several iterations can lead to additional overheads. Furthermore, the fault-tolerance issue is also not addressed in the HEED protocol.

**Maximum Energy Cluster Head (MECH):** Like TEEN and PEGASIS, Maximum Energy Cluster Head (MECH) routing protocol is proposed to improve the performance of LEACH [48]. MECH forms clusters based on the number of cluster members and transmission range. In this scheme, each node broadcasts a *hello* message to its neighbours and each node records the number of neighbours. When the number of neighbours reach to a predefined cluster number, then the node claims itself as a CH and broadcasts an advertisement to its one hop neighbours. The forwarding phase of MECH is guided by the BS to ensure multi-hop routing. MECH ensures that there will not be more than one cluster head in a certain transmission range. The main limitation of this protocol is that the initial CH selection process may select a node with low energy.

**Power Efficient and Adaptive Clustering Hierarchy (PEACH):** PEACH protocol forms clusters without any additional overhead, by using the overhearing characteristics of wireless communication [49]. By overhearing a node gets information of the source and destination of packets sent by its neighbour nodes. Overheard information is then used to form clusters. This avoids packet transmission overhead such as announcement, advertisement, scheduling messages, joining, and exchanging of control messages. Probabilistic routing protocols are used to provide adaptive multi-level clustering. A location-aware version of PEACH protocol is also proposed to minimise energy consumption by the use of global transmission schedule. For each SN, this transmission schedule is calculated during the network setup phase and the maximum transmission range of the SNs are used to avoid collisions.
Table 2.2 provides a functional comparison among three location unaware protocols. It can be seen that both HEED and LEACH incur transmission overhead since the protocols need to transmit and receive advertisements and announcements information during the CH selection and cluster formation phases. However, LU-PEACH avoids the transmission overhead by employing the packet overhearing mechanism.

<table>
<thead>
<tr>
<th>Features</th>
<th>HEED</th>
<th>LEACH</th>
<th>LU-PEACH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster information</td>
<td>Overhead</td>
<td>Overhead</td>
<td>No Overhead</td>
</tr>
<tr>
<td>Intra-cluster communication</td>
<td>TDMA</td>
<td>TDMA</td>
<td>CSMA</td>
</tr>
<tr>
<td>Inter-cluster communication</td>
<td>Multi-hop</td>
<td>One-hop</td>
<td>Multi-hop</td>
</tr>
<tr>
<td>Multi-level clustering</td>
<td>Fixed</td>
<td>Fixed</td>
<td>Adaptive</td>
</tr>
</tbody>
</table>

Table 2.3 shows the functional comparison between the location-aware PEGASIS and PEACH algorithms. The complexity of both algorithms is $O(n^2)$ and they implement the TDMA scheme for intra-cluster communication. However, in PEGASIS, the CHs directly transmit data packets to the BS, whereas LA-PEACH uses multi-hop communication to reach the BS. Furthermore, like LU-PEACH, LA-PEACH adaptively forms clusters for each packet transmission, in contrast to, fixed multi-level clustering scheme of PEGASIS.

<table>
<thead>
<tr>
<th>Features</th>
<th>PEGASIS</th>
<th>LA-PEACH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global transmission scheduling algorithm</td>
<td>Greedy TSP</td>
<td>Self-defined</td>
</tr>
<tr>
<td>Algorithm complexity</td>
<td>$O(n^2)$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Intra-cluster communication</td>
<td>TDMA</td>
<td>TDMA</td>
</tr>
<tr>
<td>Inter-cluster communication</td>
<td>One-hop</td>
<td>Multi-hop</td>
</tr>
<tr>
<td>Multi-level clustering</td>
<td>Fixed</td>
<td>Adaptive</td>
</tr>
</tbody>
</table>

PEACH significantly improves the lifetime and energy consumption of the network by using adaptive multi-level clustering and reducing transmission overheads. However, the protocol assumes that all SNs in the network have equal capabilities. Thus, PEACH is not suitable for heterogeneous WSNs.
Grid Based Data Dissemination (GBDD): In the GBDD protocol, the base station initiates the grid construction process by sending and receiving data packets [50]. The sink node is called crossing point (CP) of the grid and its geographical location \((x, y)\) becomes the starting point for the formation of grid cells. The size of the grid cell is determined by dual transmission range (RH and RL) of a sensor while working in high power radio mode and low power radio mode respectively.

The benefits of GBDD protocol is that it ensures continuous data delivery from source nodes to the base station. However, the algorithm consumes more energy when the speed is very high.

Extending Lifetime of Cluster Head (ELCH): ELCH is a hybrid protocol that combines the cluster architecture with multi-hop routing [51]. The protocol works in two phases: the setup phase and the steady-state phase. During the setup phase, the sensor nodes vote for their neighbours in order to elect CHs. Before voting, a sensor uses two rules to assess its neighbour nodes. The first rule indicates that sensors with more neighbours tend to receive more votes. On the other hand, the second rule uses a heuristic to maintain a balance in the clustering scheme. The heuristic claims that the sensor with highest proportion of residual energy to distance will gain more votes than all its neighbours. A sensor \(s_i\) uses the following equation when it votes another sensor \(s_j\):

\[
s(s_i, s_j) = \begin{cases} 
\frac{e_j}{d_{ij}} / \left( \sum_{d_{ik} \leq R} e_k / d_{ik} \right) & d_{ij} \leq R \\
0 & d_{ij} > R 
\end{cases}
\] (2.9)

Now, the total votes of sensor \(s_j\) can be calculated by adding the votes of its neighbours. Then, every sensor selects its neighbour with maximum votes as a CH and sends a message to it. In the steady-state phase, the CHs select their corresponding member nodes on the basis of their locations. After that, the time slot is assigned to each sensor node by the CH using the TDMA scheme. As soon as the clusters are formed, the CHs collect information from the member nodes and form a multi-hop routing backbone.
The ELCH protocol can minimise transmission energy and maintain a good balance in energy consumption in the network. However, if the number of members in each cluster exceeds a certain limit it will have a negative effect on the network operation.

**Scaling Hierarchical Power Efficient Routing (SHPER):** The SHPER protocol consists of two phases: initialisation and steady-state. During the initialisation phase, the base station sends request to all SNs to retrieve relative distances among them from the reply messages [52]. After that the BS randomly selects a predefined number of high and low level of CHs and broadcasts the CH-IDs of new CHs as well as the threshold level. In the steady-state phase, the most energy efficient routing paths are selected by the CHs to forward their messages to the BS.

The advantage of SHPER protocol is that energy balance is achieved, and the power depletion among the nodes is performed in a more even way. Moreover, both energy level and communication cost associated with the potential paths are considered to find a suitable route in this protocol. However, the drawback is that SHPER does not support node mobility.

**Distributed Hierarchical Agglomerative Clustering (DHAC):** First, the DHAC algorithm builds a resemblance matrix on the basis of obtained input data via HELLO messages [53]. Then, it executes the DHAC algorithm to determine the minimum cluster head for each cluster. After that control conditions are applied to cut the hierarchical cluster tree to limit the size below the predefined upper bound size of clusters. The next step is to set the lower bound of cluster size above the minimum cluster size by performing the “MERGE CLUSTERS” procedure. Finally, to select the CHs, the DHAC algorithm chooses the lower-ID node between two nodes which join the cluster first. Now, each cluster member starts to send data to CHs in turns. Here is an example of the cluster formation process run by the DHAC algorithm.

Figure 2.11(a) shows a simple network with eight sensor nodes where the dotted line represents the connectivity between two sensors. After constructing the resemblance matrix based on quantitative data, the DHAC algorithm updates the
matrix using Unweighted Pair Group Method with Arithmetic-Mean (UPGMA).
Figure 2.11(b) shows the clustering with UPGMA method and location information
using dendrogram. Then, the hierarchical cluster tree is cut to a particular level
using a pre-defined threshold value. Figure 2.12 illustrates the formatted clusters
that are constructed by cutting the hierarchical tree on the basis of transmission
range.

Figure 2.11: Dendrogram using the UPGMA with quantitative data on a sensor
network

Figure 2.12: Formatted clusters with cutting hierarchical tree
Finally, the algorithm maintains the minimum cluster size by merging two or more clusters if the size of a cluster is smaller than the predefined cluster size. Figure 2.13 shows the single node cluster \{5\} merged into cluster \{1, 2, 8\} since the minimum cluster size is set to 4.

![Figure 2.13: Formatted clusters with minimum cluster size](image)

The advantage of DHAC algorithm is that it extends the network lifetime. Moreover, the protocol ensures scalability and best routing path. However, the DHAC protocol does not support mobility, and the performance is worse when the network traffic is high.

**Energy Efficient Cluster-based Routing (EECR):** The EECR protocol is proposed to support both energy efficient routing and a wide range of network connectivity [54]. It adopts a centralised clustering approach to generate a representative path. This path is used by the sink node to select the CHs and generate the clusters. The routing procedure is divided into four phases: i) network information table generation, ii) representative paths construction, iii) cluster generation, and iv) cluster management. In the first phase, the network information table of each node is constructed using a flooding algorithm. This information is forwarded to the sink node and the sink node generates a complete network information table. In the second phase, the sink node generates representative paths on the basis of the node
connectivity and message success rate. In the third phase, CHs are selected from the representative paths and nodes are added to each cluster. Finally, to prolong the network lifetime by evenly distributing load among the sensor nodes, CHs are periodically changed during the cluster management phase. Although the protocol claims to be energy efficient and reliable, it is subject to high collisions and energy consumption due to the message flooding method used in the first phase.

Remarks: In a nutshell, the basic characteristics of the above mentioned leader-first hierarchical protocols can be summarised as shown in Table 2.4. The term ‘N/A’ denotes that the measure of the characteristic metric is not available for some protocols. However, it can be seen that most of the hierarchical protocols provide fault-tolerance, balanced clustering, and stability. Out of fourteen routing schemes, only six protocols are energy efficient, whereas seven protocols provide good levels of energy balanced routing.

Table 2.4: Comparison of the leader-first hierarchical routing protocols

<table>
<thead>
<tr>
<th>Protocols</th>
<th>Mobility</th>
<th>Overlap-ing</th>
<th>Energy Efficiency</th>
<th>Fault Tolerance</th>
<th>Balanced Clustering</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCA</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>OK</td>
<td>Moderate</td>
</tr>
<tr>
<td>TEEN</td>
<td>Fixed BS</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>OK</td>
<td>Moderate</td>
</tr>
<tr>
<td>HCC</td>
<td>Yes</td>
<td>Low</td>
<td>N/A</td>
<td>Yes</td>
<td>Good</td>
<td>Moderate</td>
</tr>
<tr>
<td>LEACH</td>
<td>Fixed BS</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>OK</td>
<td>Moderate</td>
</tr>
<tr>
<td>PEGASIS</td>
<td>Fixed BS</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>OK</td>
<td>Moderate</td>
</tr>
<tr>
<td>FLOC</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
<td>Yes</td>
<td>Good</td>
<td>High</td>
</tr>
<tr>
<td>HEED</td>
<td>Fixed BS</td>
<td>No</td>
<td>Yes</td>
<td>N/A</td>
<td>Good</td>
<td>High</td>
</tr>
<tr>
<td>MECH</td>
<td>Fixed BS</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>OK</td>
<td>Moderate</td>
</tr>
<tr>
<td>PEACH</td>
<td>Fixed BS</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Good</td>
<td>High</td>
</tr>
<tr>
<td>GBDD</td>
<td>Limited</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Moderate</td>
</tr>
<tr>
<td>SHPER</td>
<td>Fixed BS</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Very</td>
<td>High</td>
</tr>
<tr>
<td>ELCH</td>
<td>Fixed BS</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Good</td>
<td>Moderate</td>
</tr>
<tr>
<td>DHAC</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Good</td>
<td>Moderate</td>
</tr>
<tr>
<td>EECR</td>
<td>Fixed BS</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>OK</td>
<td>Moderate</td>
</tr>
</tbody>
</table>
**Cluster-first Approach:** In cluster-first approaches, all nodes first form clusters and one node from each cluster is elected as cluster head based on some specific features such as least distance and maximum power. In this clustering techniques, sensor nodes are always divided into a number of cliques in order to achieve direct communication with each other.

**Secure Distributed Clustering (SDC):** Most of the existing clustering protocols assume benign environments and cannot survive attacks from malicious nodes in hostile environments [55]. The SDC algorithm provides security mechanisms by dividing the network into multiple small groups and providing guarantee that all nodes in each small group agree on the same group membership. Figure 2.14 represents the cluster formation process for a sensor network with eight nodes. The protocol works in the following phases:

1. Each node exchanges its neighbour list with its neighbours and computes its local maximum clique.
2. Each node exchanges its local maximum clique and updates its maximum clique from the received information.
3. Each node exchanges the updated clique with its neighbours and derives its final clique.
4. Each node exchanges the final clique with its neighbours. If no clique inconsistency is detected, it terminates successfully. Otherwise, each node performs conformity checking. If it identifies malicious nodes, it removes them from the network, and restarts the protocol from first step. If malicious node is not found, it enforces the clique agreement and terminates.

The Secure Distributed Clustering protocol can survive in hostile environments since it implements security mechanisms in the cluster formation process. But it incurs communication overhead, computation overhead, and storage overhead due to the exchange of excessive messages among the nodes in the network. Moreover, SDC is not suitable for densely deployed WSNs.
Energy Efficient Geocast Algorithm (EEGA): Bomgni et al. proposed an energy efficient clique based geocast algorithm which uses the same clique formation technique described in the SDC protocol [56]. The protocol makes the assumption that the CH is located at the centre of the cluster to minimise the energy consumption. But the problem with this protocol is that if the node in central location has low energy, it will quickly lose all of its power. Thus, the protocol leads to quick network partitions.

Other Clique Based Protocols: In [57], the authors present a routing protocol for ad-hoc networks that divides the network into a number of overlapping clusters. The basic goal of this protocol is to minimise communication overhead during topology updates. In [58], the authors describe a clique based distributed group formation algorithm for autonomous agent coalitions. In this protocol, each node computes its own clique of pre-defined size and then interchanges the information to form a mutually strong inter-group communication connectivity.

Remarks: The advantage of clique based clustering approaches is that each cluster maintains a fully connected group, and thus it is easy to implement security mechanisms. But, the drawback is that the clique formation process is time consuming since each and every node in the network computes its own local clique and then interchanges the information to form the maximum global clique. In this study, we
propose a clique based clustering and routing protocol that minimises energy dissipation by reducing the clique formation processes of those nodes who are already added in clusters by some other nodes in the network.

2.3.1.3 Location-based routing protocols:

Location based routing protocols use geographical information to guide the route discovery and packet forwarding phases in WSNs. The best routing path selection, energy conservation, and the network optimisation are the primary goals of this type of protocols. In this section, we describe some common geographic based routing protocols. Furthermore, we compare the protocols to show how different routing protocols fit under different categories.

**Greedy Perimeter Stateless Routing (GPSR):** GPSR is a location based routing protocol in which nodes make local packet forwarding decisions using a greedy algorithm [59]. It makes greedy forwarding decisions using only information about a router’s immediate neighbours in the network topology as shown in Figure 2.15(a). In GPSR, packets may be trapped in holes because of obstacles or lack of adequate sensor density. In such situation, greedy forwarding is not possible and the algorithm

![Diagram](image1)

(a) Greedy forwarding algorithm

![Diagram](image2)

(b) Perimeter routing algorithm

**Figure 2.15:** Greedy and perimeter routing in GPSR
recovers this problem by routing around the perimeter of the region as depicted in Figure 2.15(b).

According to the authors, by keeping state only about the local topology, GPSR scales better than the shortest-path and ad-hoc routing protocols as the number of network destinations increases. Under frequent topology changes, the protocol can use local topology information to find a new route quickly. However, the GPSR protocol may be trapped in a blind alley if the planner sub-graph used by the GPSR’s perimeter mode is not connected.

**Geographic Adaptive Fidelity (GAF):** GAF is an energy aware location based routing algorithm designed for mobile ad-hoc networks. However, it can be adopted to WSNs as well [60]. In this protocol, each node uses the location information based on the Global Positioning System (GPS) to associate itself with a “virtual grid” as shown in Figure 2.16. The virtual grid divides the entire network area into a number of grids and the node with the highest residual energy within each grid becomes the master of that grid. Nodes in each grid are assigned specific tasks such as a node may awake for a certain period to monitor and report all events to the BS on behalf of the nodes in that particular zone. Thus, the GAF protocol conserves energy by turning off inactive nodes in the network without affecting the level of routing fidelity. Simulation results show that GAF increases the lifetime of the network by saving a significant amount of energy. The only drawback of the

![Figure 2.16: Example of virtual grid in GAF](image-url)
GAF routing protocol is that the protocol supports limited scalability and limited power management.

**Geographic and Energy Aware Routing (GEAR):** GEAR is an energy aware protocol and uses a geographically informed neighbour selection heuristics to forward a packet towards the target region [61]. In GEAR, each node keeps an estimated cost and a learned cost to reach the destination through its neighbours. The estimated cost is a combination of residual energy and distance to destination. The learned cost is a refinement of the estimated cost that accounts for routing around holes in the network. A hole occurs when a node does not have any closer neighbour to the target region than itself. If there is no holes, the estimated cost is equal to the learned cost. The learned cost is propagated one hop back every time a packet reaches the destination so that route setup for next packet can be adjusted. The GEAR protocol performs better than GPSR in terms of throughput and energy consumption during the route setup phase. However, this protocol is basically designed for mobile ad-hoc networks and does not support scalability and mobility.

**Minimum Energy Relay Routing (MERR):** The MERR protocol considers the distance between two nodes as an important routing attribute since the distance is closely related to the energy consumed on the entire path, from the source to the BS [62]. In MERR, each SN finds locally for downstream node within its maximum communication range whose distance is closest to the characteristic distance. While selecting the next hop, SN adjusts its transmission power to the lowest possible level to minimise the energy consumption. Figure 2.17 shows the selection procedure of the optimal route in MERR. The first step of this protocol (1, 2, and 3 points) is to select the relays 4, 2, and the BS. The resulting path from 5-4-2-BS approximates the optimal case and is used in the step 4 to route data from sensor 5 to the BS.

The main advantage of the MERR protocol is that it distributes the energy consumption of the sensors uniformly over the entire network. However, a major limitation of the approach is that a significant amount of energy is wasted when the sensors are very close to each other.
Hybrid Geographic Routing (HGR): The HGR protocol proposes a novel mechanism that combines both distance and direction based strategies in a flexible manner [63]. In this protocol, the main operation of a node is to define the priority \((Q_i)\) as the next hop. The greater the projected progress of node \(i\) is, the larger \(Q_i\) becomes, whereas, the lower deviation angle between the line that connects \(z\) with \(i\) and the line that connects \(z\) with \(j\) is, the larger \(Q_i\) becomes. Different forms for \(Q_i\) can be defined in order to combine both distance and direction based routing criteria.

Most Forward within Radius (MFR): MFR is a localised routing algorithm which tries to minimise the number of hops [64]. The MFR routing algorithm forwards the message to a neighbour node that makes the most progress towards the destination. Although the protocol minimises the number of hops, it does not minimise energy consumption.

Greedy and \(A^*\) Heuristic Routing (GAHR): The GAHR algorithm based on Euclidean distance is proposed for WSNs in home automation [65]. The algorithm uses a greedy forwarding technique to limit the number of hops for data transmission as shown in Figure 2.18. Furthermore, the protocol implements the \(A^*\) search algorithm to overcome unpredictable topology changes and local minimum problems.
In GAHR, every SN maintains a neighbour table that keeps a record of interfered links. When a SN detects a data transmission error, it marks the forwarding node as temporarily unstable and sets the timer. The temporarily unstable node waits to return to stable state until the timer expires. The number of recoveries from temporarily unstable state to stable state is also recorded. When the recovery count is greater than a threshold value, the corresponding link is marked as unstable. This protocol has a number of limitations. First, since the node energy is not used in the route selection procedure, low energy SNs can be selected as forwarding nodes and may quickly run out of energy. Second, the greedy forwarding selects the same
route until a node on that path exhausts its energy. Thus, the protocol results in network partitions. Third, if a SN is marked as unstable, it will never return to stable state. For example, a microwave oven is a source of radio interference in home networks. In an office, the oven is used several times to heat or cook food during office hours. Hence, the SNs near a microwave oven will be affected many times and hence may be marked as unstable. It is more likely that there will be no interference at night or on a public holiday. Therefore, the GAHR protocol is unable to deal with interference problems effectively.

Location-Based Routing Protocol (LBRP): LBRP uses location information and a greedy forwarding algorithm to route the data packets from the source node to the destination node [66]. In this protocol, some nodes records the locations of all sensor nodes in the network. These nodes are known as Location Service Node (LSN). When a node needs to send a data packet, it first triggers the Location Service Module (LSM) to obtain the location of the destination node and the next hop on the path of the destination from a LSN. Upon receiving the information, the source node sends the data packet to the next hop. The same procedure is followed by the intermediate node and the process is repeated until the packet is received by the destination node. The protocol implements a greedy forwarding approach, which selects the closest neighbour node to the destination node to forward data packets by a current node. Since the greedy forwarding approach is subject to local minimum problem, LBRP implements backtracking over multiple consecutive nodes. However, although this mechanism improves network throughput, it also incurs bandwidth overheads.

Remarks: Table 2.5 summarises important features of location-based routing protocols and also compares different routing techniques according to different attribute metrics. Most of the location-based routing protocols support limited mobility and power usage. Data aggregation and QoS are two routing attributes that are not addressed in these protocols.
Table 2.5: Classification and comparison of the location-based routing protocols

<table>
<thead>
<tr>
<th>Protocols</th>
<th>Mobility</th>
<th>Power Usage</th>
<th>Data Aggregation</th>
<th>QoS</th>
<th>State Complexity</th>
<th>Scalability</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPSR</td>
<td>Yes</td>
<td>Limited</td>
<td>No</td>
<td>No</td>
<td>Moderate</td>
<td>Very Good</td>
</tr>
<tr>
<td>GAF</td>
<td>Limited</td>
<td>Limited</td>
<td>No</td>
<td>No</td>
<td>Low</td>
<td>Good</td>
</tr>
<tr>
<td>GEAR</td>
<td>Limited</td>
<td>Limited</td>
<td>No</td>
<td>No</td>
<td>Low</td>
<td>Limited</td>
</tr>
<tr>
<td>MERR</td>
<td>Limited</td>
<td>Good</td>
<td>No</td>
<td>No</td>
<td>Low</td>
<td>Limited</td>
</tr>
<tr>
<td>HGR</td>
<td>Yes</td>
<td>Limited</td>
<td>No</td>
<td>No</td>
<td>Low</td>
<td>Good</td>
</tr>
<tr>
<td>MFR</td>
<td>No</td>
<td>N/A</td>
<td>No</td>
<td>No</td>
<td>Low</td>
<td>Limited</td>
</tr>
<tr>
<td>GAHR</td>
<td>Limited</td>
<td>Limited</td>
<td>No</td>
<td>No</td>
<td>Low</td>
<td>Limited</td>
</tr>
<tr>
<td>LBRP</td>
<td>Limited</td>
<td>Limited</td>
<td>No</td>
<td>No</td>
<td>Low</td>
<td>Limited</td>
</tr>
</tbody>
</table>

2.3.2 Multipath Routing

Although single path routing protocols are simple and scalable, they are subject to network partition, node failures, and malicious node attacks. Thus, the protocols are unable to provide communication security, even distribution of the network traffic, and reliable data transmission in WSNs. To overcome the limitations of single path routing protocols, multipath routing schemes are proposed, which select multiple paths to forward data from source to destination. Multipath routing provides a number of benefits such as load balancing, reliability, fault-tolerance, reduced delay, and bandwidth aggregation. This section summarises the advantages and disadvantages of multipath routing protocols developed for WSNs.

INtrusion-tolerant routing protocol for wireless SEnsor NetworkS (INSENS): INSENS is an intrusion-tolerant routing protocol that can defend against the compromised node attack and the DoS flooding attack [67]. INSENS uses symmetric cryptography to provide the confidentiality, integrity, and authentication. In this protocol, each node shares a secret key only with the BS. Furthermore, the routing table is constructed by the BS and disseminated to each node in the network. The protocol constructs two alternative disjoint paths between the source node and the BS. A message is sent for multiple times through each alternative path from a source to a destination.
Figure 2.19 shows routing tables built by the INSENS protocol. It can be seen that the entries in the routing table of each node have three fields: destination node, source node, and intermediate node to forward the packets. Since the protocol sends the packets towards the BS, the destination ID is zero for each node. As an example, the routing table of node 4 has the following node IDs: destination ID = 0, source ID = 5, and intermediate node ID = 1. Thus, when node 4 receives packets from node 5, it forwards the packet to node 1 since node 4 is not directly connected to the BS. In this way, each node uses its routing table to forward data packets to the next hop.

INSENS effectively defends against a number of security attacks and also minimizes the damage caused by node failures or intruders. However, the protocol consumes a significant amount of energy since it transmits multiple copies of the same message through different node-disjoint routes.

**Multipath routing (Ling et al.):** An end-to-end pairwise key establishment protocol is proposed by Ling and Znati in [68] which uses multiple disjoint paths to forward data packets. Instead of sharing common cryptographic keys with the destination node, the protocol splits the negotiated key into multiple fragments in order to secure the establishment of the common key. The destination node must
receive all fragments to rebuild the key. For forwarding data packets, the scheme uses a node disjoint routing protocol to compute the disjoint paths. However, this protocol is also vulnerable to DoS attack, selective forwarding attack, and energy exhaustive attack.

**H-SPREAD:** H-SPREAD protocol extends the previous work SPREAD that was proposed to enhance data confidentiality in ad-hoc networks [69]. H-SPREAD protocol implements the threshold secret sharing scheme \((T, N)\) to split a message into \(N\) blocks known as shares. Each share is routed via a different path towards the BS by the source node. The original packet is reconstructed by the BS if at least \(T\) shares are correctly received.

Figure 2.20 shows the path finding capability of H-SPREAD protocol, where the terms \(RPRI\) and \(RALT\) denote the primary route and alternate route respectively. The results indicate that branch-aware flooding mechanism of the protocol finds disjoint paths without any extra overhead. For example, if the average node degree is 22, the algorithm could find an average of 8 node-disjoint paths per node. Furthermore, the routing algorithm is very efficient, and the cost to find a path is lower than one message.

**Multi-Version Multi-Path (MVMP) Routing:** The MVMP routing protocol is a secure routing protocol and it ensures both security and reliability in the data communication process [70]. The protocol works in four states. First, data packets are divided into a number of groups, and different symmetric and asymmetric cryptographic algorithms are used to encrypt each group. These encrypted data packets are then reorganised into \(k\)-packet blocks and each \(k\)-packet block is converted into \(n\)-packet Reed Solomon (RS) codeword by using \(RS(n, k)\) coding technique [71]. Each codeword is forwarded to the destination node via multiple disjoint paths in order to enhance the security of the protocol. The encrypted block will be reconstructed at the receiver end after receiving at least \(k\) packets. In this protocol, authentication and integrity checks are applied by using different algorithms. However, the drawback is that the protocol does not consider resource consumption attack like DoS, replayed attack, and physical node capture attack. Thus, an adversary can
compromise a node, launch the selective forwarding attack, and even can steal the cryptographic information stored in the node.

Path Redundancy based Security Algorithm (PRSA): PRSA is a security operation support protocol that uses the Dijkstra’s algorithm to compute a least cost disjoint and braided multi-paths between communicating nodes [72]. This protocol implements a combination of transmission techniques such as round robin, redundant, and selective nodes to deliver data packets. To identify malicious nodes, the scheme uses a number of parameters like node power, node HELLO messages, and number of hops. If any malicious node is identified, then the node will be discarded from the entire network and the path list as shown in Figure 2.21. However, the protocol is subject to DoS, sybil, wormhole, sinkhole, and replayed attacks since it does not provide any preventing mechanisms.

Secure and Energy Efficient Multipath (SEEM): The Secure and Energy Efficient Multipath routing protocol for WSNs implements a modified version of the breadth first search algorithm to construct disjoint and braided paths [73].
Chapter 2. Literature Review

Figure 2.21: Flowchart of the PRSA algorithm

Flowchart:

1. Start
2. Read Network Topology
3. Find optimum routing path (Using Dijkstra Algorithm)
4. Disjoint paths?
   - Yes: Remove the nodes of the previous path
   - No: Remove every other nodes in the path and its links
5. No of routing paths < N_{max}?
   - Yes: Increment no. of paths
   - No: Select a suspected node
6. Remove the suspected nodes and its links
7. Find optimum route (Call Dijkstra routine)
8. No of routing paths < N_{max}?
   - Yes: Select a transmission mode (Selected, Redundant, Round Robin)
   - No: Stop
BS executes the route discovery, selection, and maintenance in a centralised way in order to minimise energy consumption. The authors claim that the protocol defends against network layer attacks such as replayed attack, wormhole attack, and sinkhole attack. However, SEEM does not use any cryptographic schemes, and thus the algorithm is vulnerable to a number of security attacks.

**Just Enough Redundancy Transmission (JERT):** The JERT protocol uses powerful Maximum Distance Separable (MDS) codes in order to encode the secret key [74]. More precisely saying, the secret key is encoded in \((n, k)\) MDS code by the source node and is routed through multiple multi-hop paths to the destination. In this approach, redundant symbols of the MDS codes are sent to the destination node, if the node fails to decode the secret message. The protocol sends different amounts of symbols through paths of different lengths in order to enhance the security level. However, node capturing attack is not addressed in the JERT protocol. Therefore, if a node is compromised, the adversary can delete or modify all stored information and even launch a selective forward attack. Furthermore, the protocol is also susceptible to DoS attack, network congestion, and energy exhaustion attack.

**Randomised Dispersive Route (RDR):** The RDR protocol is a randomised multi-path routing protocol for secure data collection in WSNs [75]. Instead of using pre-computed set of routes, the algorithm computes multiple paths in a randomised way when a data packet has to be sent so that the forwarding paths taken by various shares of different packets keep changing over time. Since, a large number of routes

![Figure 2.22: Randomised dispersive routing](image-url)
can be potentially generated for each source and destination, the adversary has to compromise or block all possible paths between two communicating nodes. Thus, the protocol prevents the network from compromised node attack and DoS attack. The limitation of the RDR protocol is that it does not provide defence against the rushing attack.

**Secure and Energy-efficient Disjoint Route (SEDR):** In order to maximise the network lifetime and enhance security, the SEDR protocol delivers sliced packet shares along randomly generated disjoint paths [76]. As illustrated in Figure 2.22, the protocol works in three phases: secret sharing of information, randomised propagation of each information share, and normal routing. First, packets are sliced into shares by the \((T, M)\)-threshold secret-sharing algorithm, and then SEDR scheme disperses these shares in a certain region around the source node. Second, shares are randomly forwarded along identical hop routes all over the network. Finally, the SEDR algorithm transmits shares to the sink node by using least hop routing. Although the protocol implements a new approach for secure data delivery in the network, it is vulnerable to the rushing attack and message replay attack.

**Probabilistic routing protocol for Heterogeneous network (ProHet):** ProHet uses a probabilistic strategy to determine forwarding nodes based on historical statistics of local information [77]. The protocol utilises asymmetric links to achieve assured delivery rate and improves reliability and scalability of WSNs. However, ProHet is not energy efficient and also unable to defend against security attacks.

**Secure and Reliable Multipath Routing (SRMR):** The SRMR protocol proposes a secure and reliable routing mechanism with different levels of security in an energy efficient way for WSNs [78]. The protocol splits data messages into packets that are coded using RS codes. To provide diverse levels of security, different number of fragments are encrypted related to the requested security level before being transmitted along independent node-disjoint paths. Figure 2.23 shows an example of data transmission in the SRMR protocol. First, the source node encodes each packet and generates \(M\) data fragments (i.e., 5) and \(K\) parity fragments (i.e., 3)
as a codeword of $M + K$ fragments (i.e., 8). Then, some of these fragments are encrypted depending on the required security level of an application. After that, all the fragments are sent to the sink through $n$ node-disjoint paths (i.e., 3 in this example). The sink node decrypts the packet and then decodes all fragments in order to reconstruct the original message.

\[ \sum_{i=1}^{n} x_i = M + K \]
\[ \sum_{i=1}^{n} z_i \geq M \]

**Figure 2.23:** Data transmission using eraser coding in SRMR

The SRMR protocol makes encryption feasible for energy constrained and delay sensitive WSN applications while still maintaining a robust security protection. This protocol can protect the network from compromised node attack, sinkhole attack, and wormhole attack. However, the protocol has to send 6 redundant packets for every 15 data packets and thus incurs 40% redundancy in data transmission.

**Energy-efficient ACO-based Multipath Routing (EAMR):** Energy-efficient ant colony optimisation-based multipath routing algorithm is a hybrid protocol proposed for resource constrained WSNs [79]. The route discovery phase of the protocol
is reactive, whereas it works as proactive routing algorithm after the route establishment process. When a source node \((s)\) requires to communicate with the destination node \((d)\), it broadcasts a reactive forward ant \((F_s^d)\) if the node does not have routing information regarding the destination node. Each neighbour of node \(s\) receives a copy of \(F_s^d\), say \(F_s^d.k\), where \(k\) denotes \(k\)-th replica of the broadcast message. Similarly, the next neighbour node will receive a broadcast ant, say \(F_s^d.k.l\). The task of each ant \((F_s^d.k.l..)\) is to determine a path connecting \(s\) and \(d\). Using the received information from different forward ants, the algorithm constructs multiple paths to reach the destination node.

![Forward ant](a) and Backward ant](b)

**Figure 2.24:** Paths created by forward and backward ants

**Genetic Algorithm for Energy-entropy based Multipath routing in WSNs (GAEMW):** GAEMW is a multipath routing protocol that uses the mechanisms of natural evolution to determine the routes [80]. The iterative process of the genetic algorithm is shown in Figure 2.25.

The first step of the protocol is encoding, which is performed to create the first generation. This step is known as population initialization. In this step, a routing path is encoded by a string of positive integers including all related information of the sensor nodes such as node IDs, energy levels, and distance. The second step evaluates the fitness value of the population by using the following fitness function:

\[
f(C_i) = \frac{1}{G} \sum_{k=1}^{\text{path}} p((E_{tr} + E_{rec}) \times P_{ki} \times |P_{ki}^{s,d}| - E_{avg})
\]  

(2.10)
where $G$ is the total number of paths in the network and $E_{avg}$ is the average loads of all paths. This function aims to determine the least cost multipath between the source node and the BS. In third step, a schedule builder mechanism is used to produce a solution for the scheduled tasks. When all individuals are scheduled and scored, the sum of the fitness values is calculated in order to represent the total fitness for the population. Finally, the crossover and mutation operations are performed on the individuals to generate a new population, which represents a possible solution.

The advantage of the GA-based algorithm is that it provides robustness and efficiency for global optimisation search in a complex space. However, genetic operations highly depend on randomness and thus it slows down the search process and generates low quality solutions.

**Cluster based Multipath Routing Protocol (CMRP):** CMRP is a proactive multipath routing algorithm that reduces the load of SNs by offloading most of the computation intensive tasks to the BS in the network [81]. The algorithm consists of four phases: i) neighbour discovery and topology construction, ii) cluster head
selection and cluster formation, iii) data dissemination, and iv) re-clustering and re-routing. In CMRP, the BS first broadcasts a neighbour detect ($Nbr\_DET$) message to all SNs in the network. After discovering the neighbour nodes, each node sends their neighbour information ($Nbr\_INFO$) to the BS through the relay nodes in the network. The BS receives $Nbr\_INFO$ messages from the SNs and constructs a $(n + 1) \times (n + 1)$ matrix including connectivity information of all SNs ($n$) and the BS. Based on the neighbour adjacency matrix, the BS selects the CHs and the routes to reach from the CHs to the BS.

2.4 Security Issues in Routing Protocols

The need for security in sensitive applications has lead researchers to design security extensions for both existing and new secure routing protocols in WSNs. This is because a number of attacks can be launched against the network layer in WSNs such as information disclosure, energy exhaustion, denial of service, routing table overflow, HELLO flood attack, sybil attack, blackhole attack, wormhole attack, selective forwarding, and replayed routing information attack. However, the implementation of security measures also incurs overheads that can affect the performance of the routing protocols. Therefore, a good trade-off should be maintained between cost and performance when implementing security algorithms in WSNs. The following describes the security goals, attacks and security measures used in WSNs [18].

2.4.1 Security Goals

The ultimate goal of developing security solutions for WSNs is to provide security services such as authentication, confidentiality, integrity, non-repudiation, anonymity, and availability. The following gives an overview of common security services in WSNs.

- **Availability**: Availability is concerned with the (unauthorised) upholding of resources. A variety of attacks can result in the loss of or reduction in
availability. Some of these attacks are amenable to automated countermeasures such as authentication and encryption, whereas others require some sort of action to prevent or recover from loss of availability of elements or services of a distributed system. Availability ensures the survivability of network services despite various attacks.

- **Confidentiality**: Confidentiality ensures that certain information is only readable or accessible by the authorised party. Basically, it protects data from passive attacks. Transmission of sensitive information such as military information requires confidentiality. Release of such information to enemies could have devastating consequences. Routing and packet forwarding information must also remain confidential so that the enemies could never take advantage of identifying and locating their targets in a battlefield. With respect to the release of message contents, several levels of protection can be identified.

- **Integrity**: Integrity guarantees that the authorised parties are only allowed to modify the information or message. It also ensures that a message being transmitted is never corrupted. As with confidentiality, integrity can apply to a stream of messages, a single message or selected fields within a message. But, the most useful and straightforward approach is total stream protection. A connection-oriented integrity service, one that deals with a stream of messages assures that messages are received as sent, with no duplication, insertion, modification, reordering, or replays. The destruction of data is also covered under integrity service. Thus it addresses both message stream modification and denial of service.

- **Authentication**: Authentication ensures that the access and supply of data is done only by the authorised parties. It is concerned with assuring that a communication is authentic. In the case of a single message, such as a warning or alarm signal, the function is to assure the recipient that the message is from the source that it claims to be from. Without authentication, an adversary could masquerade as a node, thus gaining unauthorised access to resource and sensitive information and interfering with the operations of the other nodes.
• **Non-repudiation**: Non-repudiation prevents either sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the message was in fact sent by the alleged sender. On the other hand, after sending a message, the sender can prove that the message was received by the alleged receiver. Non-repudiation is useful for detection and isolation of compromised nodes. When node $A$ receives an erroneous message from node $B$, non-repudiation allows $A$ to accuse $B$ using this message and to convince other nodes that $B$ is compromised.

### 2.4.2 Attacks in WSNs

Security attacks in WSNs are basically classified into four general forms: i) **outsider** vs. **insider** attack, ii) **physical** vs. **remote** attack, iii) **passive** vs. **active** attack, and iv) **laptop-class** vs. **mote-class** attack. Outsider attacks may harm the network without being a part of it, e.g., a malicious node that captures data packets. On the other hand, in an insider attack, the malicious node becomes a part of the network and performs its malicious purposes. In physical attack, nodes can be physically accessed for tampering or destroying them. In contrast, a remote attack is performed from a distance to interrupt the communication or capture data packets. In passive attack, an adversary just eavesdrops, jams or monitors the network traffic, whereas in an active attack, the adversary may modify, fabricate or suppress the data packets. Finally, in mote-class attack, the attack is implemented using the same type of hardware as sensor nodes. But in laptop-class attack, more resourceful devices in terms of computational and transmission power are used to launch security attacks.

Although security attacks can be initiated against any layer in the protocol stacks, network layer is more vulnerable compared to others. Most of the attacks on the network layer can be classified into following categories [18].

• **Energy exhaustion**– The goal of energy exhaustion attack is to waste the energy of nodes by sending unnecessary information, e.g., requesting unnecessary routes.
• **Information disclosure**– This type of attack involves disclosure of routing information by passive or active participation in the network.

• **Spoofed, altered or replayed routing information**– The aim of this attack is to change the routing behaviour by spoofing, altering, or replaying routing information.

• **Denial of Service**– DoS attack is basically attack against availability. The intention is flooding the network with unnecessary service requests like route discovery.

• **Routing table overflow**– This attack attempts to flood routing tables by creating multiple non-existing routes and thus tries to collapse the routing algorithm.

• **Hello flood attack**– The adversary intentionally injects bogus HELLO messages to remote nodes to confuse the routing protocol.

• **Sybil attack**– In this attack, a large number of pseudonymous entities are created to gain a greater influence on the network.

• **Sinkhole/Blackhole attack**– The attack aims to obtain all network packets in certain network area by representing the malicious node as attractive destination to surrounding nodes.

• **Wormhole attack**– In wormhole attack, two malicious nodes make a tunnel to give the impression to the other nodes that they have a low latency link, though, in reality, they are far away from each other.

• **Selective forwarding attack**– The compromised nodes behave selfishly, as for example, only forward particular packets in the network to save resources.

• **Rushing attack**– In rushing attack, a malicious node duplicates the original broadcast message, changes the source ID, and then rebroadcasts it to reach the SNs in distant area before the original message. If it becomes success, those SNs select the malicious node as their parent.
2.4.3 Secure Measures for WSN Routing Protocols

Several security mechanisms can be implemented in WSNs to improve the security of routing protocols. Although most of the existing protocols are adopted from other area of computer science, the special characteristics of WSNs have to be considered when implementing them in a resource constrained environment. Furthermore, the security requirements and possible attacks must be taken in consideration when selecting security measures for WSN routing protocols.

**Cryptographic Schemes:** Cryptographic mechanisms are used to provide data confidentiality, authenticity, and integrity in the network. A number of encryption schemes are developed in WSNs based on the classical cryptography. The data encryption algorithms used in sensor networks are basically classified into three major categories: symmetric-key algorithms, asymmetric-key algorithms, and hash algorithms. Hash algorithms are basically used for message authentication and require additional transmission cost [82, 83]. On the other hand, even with the use of energy efficient techniques, asymmetric cryptographic schemes consume more energy than symmetric-key algorithms [84–86]. Therefore, the symmetric-key encryption algorithms are considered more suitable to provide secure communication in WSNs. There are two types of symmetric-key encryption algorithms: i) block cipher and ii) stream cipher. Block ciphers take a number of bits as input and encrypt them as a single unit. On the other hand, each plaintext symbol (typically bytes) is encrypted one at a time with the corresponding symbol of the key-stream in stream ciphers.

The key issue in designing crypto-systems for WSNs is to maintain the trade-off among security, performance and cost. Several encryption algorithms for resource constrained SNs have been designed in the past few years. These algorithms can be classified into three main categories: compact hardware oriented cryptographic schemes, conventional block ciphers, and lightweight block ciphers. Highly optimized and compact block ciphers (e.g., KATAN and KTANTAN) are not readily suitable for WSNs, since the energy consumption and memory usage are high [87]. On the other hand, most of the classical block ciphers adopted for WSNs are vulnerable to a
number of security attacks. Therefore, the current research focuses on designing secure and lightweight block ciphers. In spite of the best efforts of researchers, many of these lightweight ciphers have relatively poor performance compared to conventional cryptographic schemes. For example, the number of CPU cycles to encrypt one byte data in conventional cryptosystems (e.g., Tiny Encryption Algorithm (TEA) and extended TEA) is less than 2000, whereas the lightweight block ciphers (e.g., LED and TWINE) require about 5500 cycles [87]. Therefore, new lightweight cryptographic schemes need to be developed to address these limitations. In this study, two lightweight symmetric block ciphers are proposed for WSNS. The algorithms are described in Section 5.5 and 5.6 in detail.

i) The Advanced Encryption System (AES) algorithm, also called the Rijndael cipher, is a widely used block cipher. AES is based on a substitution-permutation process and has a fixed block size of 128 bits. The cryptographic scheme operates on a 4 x 4 array of bytes and has a key size of 128, 192, or 256 bits with 10, 12 or 14 number of rounds respectively. Previously, a chosen-plaintext attack can break up to seven rounds of 128-bits AES and eight rounds of 192-bits and 256-bits AES [88]. Currently, AES is running on 10, 12 and 14 rounds for 128, 192 and 256-bits size key respectively and still proved breakable by researchers [89]. AES is used in many WSN protocols such as SSL and IPSEC.

ii) RC5 is a flexible block cipher that has a variable block size (32, 64, or 128 bits), number of rounds (0-255), and key size (0-2040 bits). The original encryption scheme is used a block size of 64-bits, a 128-bits key and 12-rounds. Although, RC5 is considered more appropriate for WSN applications, it requires the key schedule to be precomputed which uses 104 extra bytes of RAM per key [90]. Moreover, RC5 is patented and designed to take advantage of variable-bit rotation instruction (ROL) which is not supported by many embedded systems, for instance, Intel architecture [91].

iii) RC6 is similar to RC5 in structure and uses modular addition, data-dependent rotations, and XOR operations. It can easily be implemented in a compact form
in both software and hardware. RC6 designers claim that the cryptographic scheme was designed to be quicker than the RC5. But, the performance of RC6 has found to be approximately seven times worse than that of RC5 [92].

iv) Another commonly used encryption algorithm in WSN is Skipjack developed by the US National Security Agent (NSA). It uses an 80-bits key with 32-rounds to encrypt or decrypt 64-bits data blocks. The short key length makes Skipjack susceptible to the exhaustive key search attack [93]. An extended version, Skipjack-X is proposed by the SenSec designers to make the encryption scheme stronger against security attacks. However, the design strategy is not a proper replacement of Skipjack in WSN [94].

v) High Security and Lightweight (HIGHT) encryption algorithm is suitable for low resource devices. The algorithm suggested for 32-rounds has a 64-bits block length and 128-bits key length. Although HIGHT is designed for low-cost and low power devices, the experimental results show that it takes more memory space and operation time than RC5 [95]. Moreover, HIGHT is also vulnerable to differential attack [96], biclique attack [97], and related key attack [98].

vi) Tiny Encryption Algorithm (TEA) is notable for its simplicity of description and small memory requirement [99]. But it is susceptible to the related-key attack and chosen-plaintext attack [100]. Because of these weaknesses, a corrected block TEA, called as XXTEA, was designed to overcome the flaws in the original Block TEA. XXTEA has a key size of 128 bits and the length of the input plaintext is equal to the length of the output cipher. The last reported attack against full-round XXTEA presents a chosen-plaintext attack requiring $2^{59}$ queries and negligible work [101].

vii) DRAGON-128 and DRAGON-256 are two variants of DRAGON stream cipher [102]. Both encryption schemes generate 1088-bit state using 128-bit and 256-bit key with publicly known initialisation vector (IV) of 128-bits and 256-bits respectively. Due to large internal state and ability to authenticate short messages only, Tiny DRAGON is proposed in [103]. The authors claim that
Tiny DRAGON has greatly reduced the implementation footprint and also improved per-bit security compared to DRAGON.

viii) *Phelix* is a high-speed stream cipher which uses an 8-bit to 256-bit length key and a 128-bit IV value to generate a 32-bit output block. According to [104], Phelix is vulnerable to differential attacks. With $2^{31}$ chosen nonces, and $2^{37}$ chosen plaintext words, the key of Phelix can be recovered with about $2^{41.5}$ operations.

ix) *Salsa20* uses a hash function with 64-byte input to produce 64-byte output. It supports a key with a length from 16-byte to 32-byte and an IV of length 16 byte. The algorithm encrypts a 64-byte block of plaintext by hashing the key (128-bit), nonce (64-bit) and sequence number (64-bit) and xoring the result with the plaintext. A powerful attack has been proposed by researchers in [105] against Salsa20. This attack is a related cipher attack that claims to recover the 256-bit secret key with the computational complexity of about $2^{224}$.

x) *KATAN* and *KTANTAN* are two block ciphers proposed by Canniere et al. [106]. Both ciphers use blocks of sizes 32, 48 or 64 bits under 80 bits key and iterate for 254 rounds. The main difference between KATAN and KTANTAN is the key scheduling scheme. The 80 bits key in KATAN is loaded into a register and is repeatedly clocked, whereas in KTANTAN the key is fixed. These two encryption schemes are vulnerable to a number of security attacks. A conditional differential cryptanalysis with a practical complexity in a single key settings as well as related key settings against KATAN presented in [107, 108]. Similarly, a meet-in-the-middle attack is proposed against KTANTAN in [109] that recovers the 80-bits secret key of the full rounds KTANTAN (32/48/64 bits) at time complexity of $2^{72.9}$, $2^{73.8}$ and $2^{74.4}$.

xi) *LED* is a lightweight block cipher that encrypts 64 bits blocks using either 64 bits or 128 bits size of key with 32 or 48 rounds respectively [110]. Instead of key scheduling, the key is xored for every four rounds in LED. This feature is compensated by an increased number of rounds when compared with the AES.
The authors in [111] present a meet-in-the-middle attack against 8 rounds of LED-64 and 16 rounds of LED-128 with slightly lower complexities compared to the exhaustive key search. In [112], the authors present results concerning differential cryptanalysis of LED and an observation on full LED in the related key settings.

xii) TWINE is a lightweight 64 bits block cipher that uses an 80 bits or a 128 bits key [113]. It employs a generalised feistel structure with 16 branches and iterates for 36 rounds. The internal F-function is repeated 8 times in each round and is composed of a sub-key addition and a single S-box. The best known attacks against TWINE are two biclique attacks on TWINE-80 and TWINE-128 with the time complexities equal to $2^{79.1}$ and $2^{126.8}$ respectively with a data requirement for the two attacks equal to $2^{60}$ [114].

xiii) A chaos block cipher with integer parameters has been proposed in [115]. The algorithm divides the plaintext into 8-bit blocks and then performs the bit permutation. The permuted bits are encrypted in four rounds Feistel cipher using four bytes sub-keys. These sub-keys are generated by performing XOR operations with 32-bits integer chaos. Finally, the 8-bits are permuted again to generate the corresponding ciphertext. However, this chaos block cipher cannot resist the differential attack because the number of rounds is too small and the calculation precision of round function is too short [116]. Moreover, the energy consumption and memory usage by the cipher are also not analysed.

xiv) Another block cipher based on chaotic S-boxes and a substitution-permutation network was proposed in [117]. The encryption procedure avoids floating-point operations and multiplications in order to minimise the energy consumption. Similar to RC5, the block cipher based on chaos (BCC) supports variable word size ($w$), number of rounds ($r$), and length of the encryption key ($b$). Although the authors claim that BCC has good diffusion property and low energy consumption, they didn’t provide any security and performance analysis.
xv) Xiao-Jun et al. proposed a fast, secure, and low resource consumption algorithm based on the integer discretisation of a chaotic map, the Feistel network structure, and an S-box [118]. The encryption algorithm uses a block length of 32 bits, a key length of 128 bits, an initial vector of 32 bits, and 14 rounds iteration to generate the 32 bits ciphertext. Experimental outcomes and analysis show that the cipher has a large key space, very good diffusion, and good statistical balance. The main drawback of this algorithm is that 32-bit block size is not semantically secure against a chosen-plaintext attack.

xvi) Elliptic Curve Cryptography (ECC) is most commonly used for public key cryptography. As for example, the hardware implementation of elliptic curve based asymmetric cryptosystem for WSNs is proposed in [119]. The paper presents a 160-bit ECC processor over prime field that meets the security requirements for sensor networks. The authors in [120] proposed the TinyECC which implements different combinations of optimisations in order to provide greater flexibility in integrating into sensor networks. In addition to encryption, ECC is also used for pre-distributing secure keys among the sensor nodes [121]. In the secure key pre-distribution scheme, each sensor is assigned a seed key prior to its deployment which is a distinct point in the elliptic curve. Then, each sensor node generates its private key ring using point doubling operation over the seed key. Two nodes are paired with each other and start communication when both of them share a common private key. However, elliptic curve operations (e.g., scalar multiplication) are highly expensive in terms of CPU consumption.

For security applications in WSNs, best cryptographic scheme selection in terms of energy efficiency and small memory is a challenge. The table 2.6 shows the benchmarked performance of different lightweight block ciphers in the context of small embedded platforms [87]. It can be seen that TWINE, XTEA, HIGHT, and AES have good performance in terms of the CPU elapsed time for encryption and decryption procedures. Hardware-based cryptographic schemes such as LED, KATAN, and KTANTAN have poor results compared to the software-based schemes. TEA,
XTEA, TWINE, and LED use relatively low memory space than the other encryption schemes.

**Table 2.6: Comparison among different lightweight cryptographic algorithms**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Block Size (bits)</th>
<th>Encryption (cycles)</th>
<th>Decryption (cycles)</th>
<th>RAM Usage (bytes)</th>
<th>ROM Usage (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>128</td>
<td>30257</td>
<td>38508</td>
<td>19</td>
<td>4460</td>
</tr>
<tr>
<td>HIGHT</td>
<td>64</td>
<td>32372</td>
<td>32623</td>
<td>18</td>
<td>3130</td>
</tr>
<tr>
<td>SKIPJACK</td>
<td>64</td>
<td>84923</td>
<td>123362</td>
<td>19</td>
<td>6628</td>
</tr>
<tr>
<td>TEA</td>
<td>64</td>
<td>8725</td>
<td>9129</td>
<td>13</td>
<td>1354</td>
</tr>
<tr>
<td>XTEA</td>
<td>64</td>
<td>9287</td>
<td>9613</td>
<td>11</td>
<td>1395</td>
</tr>
<tr>
<td>KATAN</td>
<td>64</td>
<td>1518391</td>
<td>1397924</td>
<td>1953</td>
<td>8348</td>
</tr>
<tr>
<td>KTANTAN</td>
<td>64</td>
<td>11004783</td>
<td>10864265</td>
<td>790</td>
<td>6252</td>
</tr>
<tr>
<td>LED</td>
<td>64</td>
<td>894680</td>
<td>897352</td>
<td>41</td>
<td>2670</td>
</tr>
<tr>
<td>TWINE</td>
<td>128</td>
<td>82003</td>
<td>60932</td>
<td>23</td>
<td>2216</td>
</tr>
</tbody>
</table>

**Key Establishment Schemes:** Almost all cryptographic schemes involve some sort of key exchange procedures. WSNs require efficient and secure key distribution protocols since nodes are randomly deployed in a large geographical area. Due to the limited computational power and memory of the sensor nodes, the well-known key exchange protocols, such as Diffie-Hellman key exchange protocol, could not often be used in WSNs. Recent researches have proposed various key establishing protocols like Peer Intermediaries for Key Establishment (PIKE), Hierarchical Key Establishment (HIKE), Random Perturbation Based (RPB) scheme, and Constant Random Perturbation Vector-based (CRPV) scheme [122–124]. However, these key establishment schemes are implemented independently irrespective to encryption procedure. Thus, it is necessary to design key generation and exchange mechanisms that uses the same security architecture of encryption algorithms.

**Trust and Reputation:** Trust and reputation metrics are used to verify compromised nodes or node failures in the network. Since, the huge number of sensor nodes can cooperate in a collaborative manner in the network, it is very hard to monitor malicious behaviours of individual node. Hence, either a centralised or decentralised trust and reputation management scheme has to be used to exchange trust values
among the nodes. Based on the current value of trust and reputation metrics, more trustworthy routes could be found or misbehaving nodes could be isolated from the network. However, implementation of these mechanisms incurs cost since it requires to maintain and exchange the trust or reputation scores periodically. Figure 2.26 shows the structure of a generic trust entity designed for sensor networks.

In this trust entity, the main function of information module is to collect information about the behaviour of the neighbour nodes. Information can be perceived either through observations and experiences, known as first-hand information, or by sharing the observed events with other nodes, known as second-hand information. Upon receiving the information, the reputation manager module can use this list of events to calculate and store the reputation of its neighbour nodes. Finally, this reputation metric will be used by the trust manager module to measure the trust values. The trust value can be used to select best nodes that can be trusted for particular communication and also to detect the nodes behaving maliciously. Here, we describe the features of a trust management system for WSNs.

Information Gathering: Behavioural-based trust management system requires to collect information about the behaviour of the nodes in the network. This information gathering process can be done in several ways such as centralised, distributed, and hybrid way. In centralised system, the BS is completely responsible for collecting, maintaining, and distributing the information. On the other hand, in a distributed system, each node has to keep its own measurement on collected information. The hybrid information gathering system implements a distributed process among the nodes in each cluster and a centralised process between the CH and the BS. Once, the network starts functioning, the nature of network traffic and communication behaviour will determine how to calculate the reputation and trust values of a node. Some common features that are used to identify misbehaving nodes are frequent packet drops, inactive for a long period of time, constantly switching between active and inactive states.

Sources of Information: In addition to protocol-specific events, there is a large set of generic events that can be used as inputs for a trust management systems in WSN.
For example, hardware related errors, deviations from sensor readings, number of repeated messages can be useful to model the behaviour of a certain node. From the number and format of the messages sent inside the network, it might be possible to identify the existence of repeated or malformed packets, the selective delaying or dropping of packets. The existence of repeated packets indicates the possibility of problems in communication channel. Similarly, repeated malformed packets (e.g., with an invalid integrity code) indicate the presence of a malicious node. Delaying the routing process can also be detected from the network environment by comparing the time consumed on forwarding an incoming message with the average time of the network.

Designing Computation Model: Upon receiving the collected information, the trust entity is able to calculate the trust measurements with the help of the reputation manager. Due to the resource constrained nature of sensor nodes, it is not possible for a sensor node to store all events generated by its neighbours during its lifetime. Hence, it is necessary to design a lightweight reputation manager that could capture and efficiently store the behaviour of other entities. The other part of the trust entity is the trust manager that calculates trust measurement of a node using existing reputation as an input and helps to make a decision regarding the node behaviour.

Chen et al. [125] proposed a trust management system based on statistical and
mathematical analysis. The authors build up a reputation space and trust space to define a corresponding transformation from the reputation space to the trust space. Moreover, some important properties about the relationship among positive outcomes, negative outcomes, total outcomes, positive trust, negative trust, and certainty are also discussed. However, the model is rather theoretical, and there is no implementation and therefore, no experimental results.

An agent-based trust and reputation management scheme (ATRM) for clustered WSNs with backbone is proposed in [126]. In this system, each node holds a mobile agent which is responsible of monitoring and administrating the nodes trust and reputation. Before a transaction between two nodes can take place, the mobile agent of the requester is sent to the provider node to obtain a certificate. Based on this certificate a decision is made whether the transaction will take place. As soon as the transaction is completed, the requester creates an evaluation report regarding the quality of service for the transaction. This evaluation report is then sent via the requesters mobile agent to the providers mobile agent. A mobile agent periodically issues an updated certificate based on these collected evaluations. The experimental results show that ATRM has low latency and requires minimal overhead.

Chen et al. proposed a new protocol ETSN that contains two types of nodes: the agent nodes and the sensor nodes [127]. Each agent node is responsible to monitor the behaviour of sensor nodes within its radio range. Based on the monitored data, the agent node computes the trust rating and then, broadcasts it. All other sensor nodes which receive the trust rating, immediately update their trust rating values. On the basis of the updated trust rating, a decision is made whether to cooperate a certain sensor node or not. The experimental results and analysis show that the ETSN protocol is robust in detecting malicious nodes and scales well. Furthermore, compared to the other examined schemes (ATSN [128] and RFSN [129]), ETSN is more suitable for WSNs.

In [130], the authors proposed a Generic Communication Protocol (GCP) that can be used to exchange the trust values. In GCP, a theoretical energy consumption analysis and evaluation of three state-of-the-art reputation-based trust management
schemes (GTMS [131], RFSN [129] and PLUS [132]) for WSNs are also presented. The simulation results show that GTMS dissipates low energy compared to PLUS and RFSN for the tested peer recommendation scenario.

Zahariadis et al. proposed an Ambient Trust Sensor Routing (ATSR) which relies on a distributed trust model considering direct and indirect trust [133]. ATSR adapts the location-based routing principle to cope with large network dimensions. The experimental results show that significant energy is consumed for routing and trust purposes, and thus the frequency of exchange of this information should be very well considered.

There are several trust management systems starting from using mobile agents to transport trust rating, to sharing trust and reputation information to local neighbourhood as well as to the BS. However, several colluding compromised nodes in WSNs still can manipulate the system by praising untrustworthy nodes. Furthermore, the decision making process involves different mechanisms from simple majority schemes to complex statistical schemes, and thus offers several opportunities. Finally, the reactive mechanism against misbehaving node is also a critical issue. For example, excluding a node from the network should be done carefully so that the mistakenly exclusion of nodes due to temporary environmental interferences can be avoided. All these issues have to be considered to design an effective trust and reputation management system.

**Secure Localisation:** Securing the localisation of sensor nodes helps each node to determine its position in location-based routing algorithms and thus prevents the malicious node from claiming itself as legitimate one. This security measure can be used against wormhole attack and sybil attack.

### 2.5 Research Challenges

Since limited battery power is the main source of energy in WSNs, it is highly recommended that WSN protocols should function in an efficient way in order to extend the lifetime as much as possible. Hierarchical and geographical single path
routing is more energy efficient than flat-based single path routing and extensive efforts have been exerted to design efficient single path routing protocols in past decades. In spite of all efforts, still there are some challenges that confront effective solutions of the routing problems. For example, in location-based single path routing, many protocols use heuristic functions to guide the route finding procedure in the right direction. The heuristic function is a good choice due to its capability to find (almost) optimal path in polynomial time for a large scale sensor network. However, most of the algorithms use greedy search techniques on the basis of the heuristic values obtained by the heuristic functions. As a result, the algorithms may not find the optimal routes and thus the data packets are required to traverse long path to reach the destination node. Furthermore, energy is not considered as one of the route selection metrics in many routing protocols. Therefore, some of the forwarding nodes quickly exhaust their energy and result in network partitions. To overcome these problems, an energy aware heuristic protocol with optimal routing has to be designed for sensor networks.

Clique-based hierarchical single path routing protocols maintain a fully connected graphs, and thus facilitate the implementation of security mechanisms. However, the clique formation process is time consuming since all nodes have to compute their own cliques and then need to exchange the information with neighbour nodes to form the maximum global cliques. Thus, clique-based protocols result in communication, computation, and storage overheads. Moreover, initiating cluster formation process in every round leads to unnecessary energy consumption. Therefore, optimisation of the clique formation process as well as the elimination of unnecessary clustering processes is necessary to improve the energy efficiency of clique-based hierarchical single path routing algorithms.

Although single path routing protocols are suitable for many WSN applications, they are subject to a number of security threats. Single path routing has to implement several security measures to ensure secure data communication in the network. However, security itself contributes to the energy consumption due to extra communication and processing done by each sensor node. Even worse, energy is highly
affected when both security mechanism and multi-path routing are combined together. Hence, designing lightweight and robust security protocols is a challenging task. Elliptic Curve Cryptography (ECC) is becoming more popular in WSNs for its smaller size of key compared to other public key cryptographic schemes. But, elliptic curve operations (e.g., scalar multiplication) are highly expensive in terms of CPU consumption. However, genetic operations and chaotic maps can be combined with elliptic curve operations (over prime field) to exploit the advantages of both asymmetric and symmetric encryptions.

Due to the complexity and variety of security solutions, it is not possible to design a single solution that can achieve all security goals. Rather, depending on the applications, security measures have to be carefully chosen to maintain a balance between the security level and minimal utilisation of available resources. To establish a secure communication channel between the source node and the destination node, the security measures must be tightly coupled with routing protocols since the implementation gap may lead to security hole in the network. Thus, it is a challenge to develop a common security framework and integrate the security algorithms with routing protocols.

The implementation of cryptographic schemes on single path routing can defend against most of the security threats. However, the security measures are unable to protect the routing protocols from node failures, jamming node attack, and compromised node attack. Since the protocols use a single path to send data packets, if a node is compromised then all packets going through that node can be captured by the attacker. To overcome this problem, secure multipath routing protocols have to be designed for WSNs. Multipath routing schemes can easily overcome node failures and jamming node attack by using multiple node disjoint paths to forward data packets. However, to minimise the effects of compromised node attack, a node monitoring system or trust based system has to be designed to work together with a multipath routing protocol. Integration of different security measures and a node monitoring system with an energy efficient multipath routing protocol is a challenging task.
Research methodology can be described as a sequence of steps for systematically analysing scientific questions, designing and executing experiments to answer those questions, and validating the outcomes. It consists of a number of activities: i) observation: is an act of recognising a fact or occurrence, ii) hypothesis: can be defined as a tentative assumption in order to test the logical or empirical consequences, iii) evaluation: is conducted to test the model as a means of analysis, and iv) prediction: involves the action of predicting future events. This chapter describes research methodologies, research questions, and hypotheses in detail.

3.1 Research Questions

From the limitations and research challenges discussed in the previous chapter, a number of research questions are identified which need to be addressed for efficient and secure communication in WSNs. The following research questions motivated us to design and develop new protocols for sensor networks.

*RQ-1.* How can energy efficiency of clique-based hierarchical routing protocols be improved through optimisation of the clique formation process?

*RQ-2.* How can an energy efficient location-based routing protocol be designed using the heuristic function and heuristic search to prolong the network lifetime?
RQ-3. How can lightweight and secure cryptographic schemes be designed using elliptic curve and genetic operations based mechanisms in WSNs?

   a) How can security measures be combined with routing protocols to defend against security threats?

RQ-4. What are the effective ways to tightly couple security mechanisms and routing protocols to establish a secure communication channel between a source and a destination node?

### 3.2 Hypotheses

The research questions mentioned above are addressed by the following hypotheses.

1. Initiating the clique formation process on the basis of the degree of connectivity can help to minimise energy dissipation in clique-based protocols. Moreover, implementation of a threshold limit on the energy level of SNs can help in reducing the unnecessary cluster formation process in each round and thus can save a significant amount of energy (RQ–1).

2. Routing protocols can be designed by combining both local and global information as a foundation to prolong the network lifetime (RQ–2). An energy and interference aware heuristic function can be implemented to construct alternative routes to avoid the low energy sensor nodes as well as the interfered links.

3. Simple elliptic curve-based encryption algorithms can be developed for resource constrained sensor nodes. Elliptic curves and chaotic maps can be combined with genetic operations to provide an enhanced level of security in WSNs (RQ–3). Furthermore, multipath routing protocols can be designed since they can effectively handle node failures, jamming node attack, and compromised node attack (RQ–3a).
4. One possible solution for RQ–4 can be achieved by developing a common security framework for different security measures such as MAC and OHC generation, authentication, and encryption procedures. This framework has to be integrated with routing protocols to provide security in both route construction and data transmission phase.

### 3.3 Research Methodology

The schematic view of the research process followed in this thesis is shown in Figure 3.1. Research methodology is divided into three major phases that are further subdivided into a number of stages.

The first phase is the *investigation phase*, where WSN characteristics and environments, security, and communication protocols are reviewed, then the limitations of existing protocols are identified. Problem domain and research scopes are also identified in this stage. The second phase is the *analysis and development phase*,

![Figure 3.1: Research methodology](image-url)
where comparative analysis is conducted on several protocols used in different applications. Based on the comparative analysis, either existing protocols are modified or new protocols are suggested. This phase proposes new protocols that are suitable for small to large scale WSN applications. The final stage of research methodology is the **evaluation phase**. This phase involves the implementation and testing of the proposed protocols. Evaluation outcomes validate the efficiency and accuracy of the proposed protocols.

### 3.3.1 Investigation Phase

In this phase, a comprehensive study on WSN characteristics, deployment environments, and application domains is conducted. Further, the existing literatures on communication protocols and security algorithms in WSNs are investigated. Then, the research questions and hypotheses are identified on the basis of the literature survey on existing mechanisms. All steps involved in the investigation phase are illustrated in Figure 3.2.

### 3.3.2 Analysis and Development Phase

This phase is divided into three sub-phases, namely methods for analysis, conceptual model design, and developing the proposed model. The knowledge and facts obtained in the previous stage are used to analyse and investigate whether they support the design rational or not. Several methods such as cause and effect analysis, theoretical analysis, statistical analysis, and SWOT (Strength, Weakness, Opportunities, Threats) analysis are used to compare the efficiency and security features of different protocols. The analysis leads to the design of a conceptual framework to assess the applicability of the new model. If the model is not feasible then it returns to the initial stage to exert a new model. Otherwise, it directs to the development stage.

Most of the hierarchical routing schemes consume a significant amount of energy in the cluster formation and maintenance process. Periodical re-clustering and
rotation of CHs in each round are mainly responsible for this energy dissipation. However, investigation shows that the use of the threshold level can remarkably improve the network lifetime in WSNs. Another problem in clique-based hierarchical routing is that the clustering and routing approaches initiate repetitive clique formation processes and result in computational, communication, and storage overheads. To overcome these problems, a new clique based clustering and routing scheme is developed which avoids unnecessary clique formation processes and thus saves a significant amount of energy. Development of this protocol addresses RQ-1.

The literature survey shows that hierarchical and location-based routing is more suitable than flat-based routing in WSNs. Therefore, in addition to the hierarchical protocol, a location-based interference aware adaptive routing protocol is developed.
to address RQ-2. The protocol is suitable for small-scale networks such as home or office automation networks. In this protocol, a heuristic function and the A* search algorithm are used to find an efficient route from a source to a destination node by avoiding low energy nodes and interfered links. The main design objective of this protocol is to extend the network lifetime through delaying network partitions.

The RQ-3 asks, “How can lightweight and secure cryptographic schemes be designed using elliptic curve and genetic operations based mechanisms in WSNs?” Since the implementation of security mechanisms incurs cost, it is a big challenge to design energy efficient security mechanisms for WSNs. There must be a trade-off between the cost and the level of security provided. A simple lightweight encryption algorithm based on the pseudorandom bit sequences generated by elliptic curve operations can be designed for sensor networks. In addition, a chaotic map and genetic operations can be combined with standard cryptographic schemes to provide an enhanced level of security. An energy efficient secure multipath routing protocol is designed and integrated with the proposed security algorithms (RQ–3).

The RQ-4 asks, “What are the effective ways to tightly couple security mechanisms and routing protocols to establish a secure communication channel between source and destination node?” WSNs use different mechanisms for key distribution, authentication, and encryption procedures since there is no common security framework. However, the implementation of individual security mechanism in sensor
networks may result in an implementation gap that can be exploited by adversaries. Therefore, a common security framework should be designed to minimise the gap. For example, the same seed key can be used in both the node verification and encryption process so as to derive a new session key. To address RQ–4, a chaotic map and genetic operations based cryptographic scheme can be used to implement different security measures and can also be integrated with a multipath routing protocol.

3.3.3 Evaluation Phase

Evaluation phase involves four major actions: implement, evaluate, modify, and test. After implementation, the model is evaluated and compared to see whether the design goals are achieved or not. If the model is unable to fulfil the design objectives, it has to enter the modification and testing procedure. This process continues until the desired design goals have been achieved.

The performance evaluation of routing protocols and security mechanisms are generally based on energy consumption, network lifetime, memory requirements, computational complexity, and communication cost. An analytical model is designed for lifetime estimation of WSNs. This model is used to formally define the lifetime of single path and multipath routing protocols. In addition, both virtual and real sensor motes have been used to implement the proposed protocols. The Mica2 and Arduino Pro sensor motes are used as hardware platforms to run the experiments. The simulation software MATLAB, OMNeT++, TOSSIM, and PowerTOSSIM are used to calculate the cluster size, coefficient of variance, number of single nodes, energy dissipation, memory use, network lifetime, encryption, and communication time. On the other hand, the emulator ATEMU is used to measure the elapsed CPU cycles for security algorithms. The protocols are implemented in C++ and nesC programming platforms.

For quantitative analysis, a number of performance metrics are compared in the evaluation phase. Average cluster size, coefficient of variance, and number of single node clusters are measured using the MATLAB simulator. OMNeT++ is also used to compute the network lifetime, energy dissipation, communication delay, control
overhead, and throughput of routing protocols [134]. Memory is a scarce resource in sensors, and a protocol using less memory has an advantage as it allows an application to use free memory. Memory occupation is calculated using TOSSIM [135] for a number of security measures.

The number of instructions used by a routing scheme is also important in designing efficient routing protocols. ATEMU is used to compute the number of instructions on a Mica2 mote [136], whereas encryption and communication time is measured by using PowerTOSSIM [137]. For Arduino Pro mote, the Arduino IDE is used to obtain the CPU cycles, encryption, and communication time by using a serial interface [138].

In addition to performance measurements, a number of block ciphers are evaluated in real sensor motes to investigate their applicabilities in WSNs. Some critical insights are also provided which suggest to maintain a trade-off between cost and security level. Furthermore, the security of the proposed multipath routing protocol is
evaluated in the presence of adversaries. A number of security attacks are designed to test the reliability and effectiveness of the proposed scheme.

3.4 Summary

This chapter describes the research questions, hypotheses, and research methodology in detail. A systematic research process is developed to solve the research problems. We have employed the best research standards available, including analytical and quantitative analysis for numerous performance and security metrics. The next three chapters of this report propose energy efficient routing protocols and lightweight security measures developed to achieve the desired research goals.
This chapter proposes energy efficient single path routing protocols for different types of WSNs. The Clique-Based Clustering and Routing (CBCR) is an energy efficient protocol for large-scale WSNs, whereas Interference Aware Heuristic Routing (IAHR) is designed for small-scale sensor networks. Section 4.2 describes the CBCR protocol, which groups sensor nodes in a number of clusters and selects a cluster head for each group. The cluster head receives data packets from its member nodes and forwards them to the base station via other cluster heads. The protocol extends network lifetime by optimising clique formation process, hierarchical forwarding, sleep scheduling, and load balancing. On the other hand, Section 4.3 describes the IAHR protocol, which implements a heuristic search algorithm to find efficient routes. It improves network performance by the use of local and global topology information, remaining energy, and link quality. The proposed protocols are tested in a simulator and a detailed analysis of results is also provided.

4.1 Introduction

The low overhead of route discovery and maintenance makes single path routing an appealing research area in WSNs. Furthermore, it offers significant advantages (e.g., scalability) over multipath routing for many applications such as wireless mesh networks (WMNs), and wireless home automation networks (WHANs). WMNs
implement single path routing since reliability is not affected because of redundant connectivities in the network. Similarly, a small-scale network such as WHAN is ideal for the implementation of single path routing since it is hard to construct multiple node disjoint routes due to the small number of SNs in the network. However, some challenges need to be addressed to ensure energy efficiency and reliability. Routing protocols for large-scale WSNs should be energy efficient since SNs may be deployed in a remote area, and thus it can be difficult to replace or recharge them. On the other hand, small-scale WSNs routing protocols should effectively handle link failures and interference problems to ensure reliable data transmission. Therefore, different WSN routing protocols should be designed to cope with different network scenarios. Due to self-organisation and hierarchical forwarding, the CBCR protocol is suitable for WSNs with densely located SNs in remote area. Furthermore, it implements the TDMA-based sleep scheduling mechanism to avoid simultaneous intra-cluster data transmission. Thus, SNs send information during their pre-assigned TDMA slots and then switch to sleep mode in order to conserve energy. On the other hand, nodes are always active in the IAHR protocol since they continuously collect information from the environment and route data packets to the BS. Small-scale networks such as home networks are subject to high interference and frequent topology changes due to the presence of various radio enabled and mobile devices. Thus, link quality and node mobility are two critical factors which must be considered in the design of routing protocols. The self-adaptive characteristic of IAHR protocol makes it more suitable for dynamic environments.

4.2 Clique-Based Clustering and Routing Scheme

The CBCR protocol is suitable for large-scale sensor networks such as gas-field monitoring, industrial control, and habitat monitoring. The protocol implements TDMA-based transmission scheduling scheme which enables node to power off and switch to low energy sleeping mode during idle period. Since CSMA (Carrier Sense Multiple Access) can lead to highly congested networks due to collisions in high traffic scenario, TDMA provides a much more energy reliable mechanism for passive
monitoring in WSNs. Furthermore, CSMA loses efficiency as the radio unit needs to be kept on even when the node is not sending any information [139]. In TDMA, each node is assigned a given time slot to freely transmit without interfering other nodes and thus the approach ensures better energy consumption compared to CSMA. Figure 4.1 shows a wireless mesh network built for remote sensing and control of a large area. In this network, the sensor nodes at the core of the network forward sensed information to the BS in a multi-hop fashion. The BS is connected to the Internet which facilitates the end user(s) to monitor and control the network. The versatility and scalability of WMNs allow it to be involved in a large number of applications such as environmental monitoring, security and surveillance systems, health and medical systems, building automation, and transport systems.

CBCR is a clustering based protocol which forms non-overlapping clusters of size \( n \); where \( n \) is the maximum cluster size. By exchanging information of 1-hop neighbours, all SNs in the network are grouped into a number of disjoint cliques, in which all sensor nodes can directly communicate with each other. Among all nodes in a cluster, the node with the maximum energy becomes the cluster head.
Chapter 4. Energy Efficient Single Path Routing

The key features of the CBCR protocol are: self-configuration and localised coordination, maximum energy cluster head selection, rotation of cluster head on the basis of threshold value, hierarchical forwarding, load balancing, fault tolerance, and scalability. The details of the protocol is described below.

4.2.1 Network Model

It is assumed that a WSN is comprised of several hundred nodes. Each SN supports different levels of transmission power and can dynamically adjust the power level. As for example, the TelosB mote equipped with a CC2420 radio module enables eight different transmission power levels that can be dynamically adjusted using Adaptive Transmission Power Control schemes [140, 141]. Further, the following assumptions are considered to define the network model.

- Every SN is assigned a unique node ID.
- Links are bidirectional, i.e., two nodes $u_1$ and $u_2$ are connected if both of them can communicate with each other using the same transmission power.
- SNs are equipped with limited and non-chargeable batteries.
- The network consists of multiple mobile/stationary nodes, which implies that energy consumption is not uniform for all nodes.
- Network topology does not change during the cluster formation process.

4.2.2 The CBCR Protocol

The CBCR protocol is divided into four phases: cluster formation, setup, steady, and forwarding. Following is the detailed explanation of the proposed protocol.

1. Cluster Formation Phase: In this phase, each node receives the connectivity lists from its 1-hop neighbours and computes the degree of connection from the lists. In addition to this, the received signal strength of each neighbour is also measured
and saved initially. The connectivity matrix formation process by node \( m \) is presented in Algorithm 4.1. First, node \( m \) receives node lists from its neighbours and finds out the common connections between each of its neighbours and itself. Then, it forms the connectivity matrix including the common connections of neighbour nodes i.e., the degree of connectivity and the energy level. In this way, each node constructs connectivity lists before starting the clique formation process. When the matrix is constructed, each node checks whether it holds the highest degree of connection. If not, then it waits until it is added to a cluster by its neighbour nodes or it becomes the highest degree node after some iterations. Otherwise, it waits for a random period of time \( t \) and initiates the cluster formation process.

**Algorithm 4.1:** Formation of Connectivity Matrix

**INPUT:** \( \text{NNL}_i \); \( \text{NNL} \) denotes neighbour node list of node \( i \), where \( i \) is the neighbour of node \( m \)

**OUTPUT:** \( G_m \); Connectivity matrix of node \( m \)

**STEPS:**

\[
\text{DoC}_i = \text{NoE} (\text{NNL}_i); \quad \text{Degree of connectivity equals to elements in NNL}
\]

for each neighbour node (\( i \)) of node \( m \) do

\[
\text{for } k = 1 \text{ to } \text{DoC}_m \text{ do}
\]

\[
\text{if } \text{NLL}_{i,k} == \text{NLL}_{m,k} \text{ then}
\]

\[
\text{Add NLL}_{i,k} \text{ to } G_m
\]

end

end

Add \( \text{DoC}_i \) and \( \text{RSS}_i \) to \( G_m \)

Maximum clique computation in a random graph is an NP-complete problem [142]. Therefore, a greedy algorithm is used to compute maximal cliques as shown in Algorithm 4.2 to reduce the time complexity [143]. It is assumed that node \( m \) holds the highest degree of connections among all its neighbour nodes, and thus the algorithm is executed by node \( m \). When node \( m \) forms the cluster, it selects the cluster head and announces the information to its neighbour nodes. The information includes the node ID of all cluster members followed by the cluster head node ID.
Algorithm 4.2: Cluster Formation Process

INPUT: $G_m = \{V_i, E_i\}, i \in V_i$ ; Connectivity matrix of node $m$

OUTPUT: $C_m$ ; Cluster node list, initially holds $m$

STEPS:

$CS = n$; ; Set cluster size, it equals to node size by default

$CN = m$; ; Set current node to $m$

if $G_m$ of node $m$ is null then
  return ; indicates single node cluster
else
  while (size of $C_m \leq CS$ and exists nodes in common) do
    Find $k$, having maximum common connection with $CN$
    Add $k$ to $C_m$
    $CN = k$;
  end
end

Algorithm 4.3 shows the cluster head selection and announcement process initiated by node $m$. After receiving the cluster information sent by node $m$, all neighbour nodes of node $m$ will inform their $1$-hop neighbours regarding their memberships. Hence, the non-clustered nodes will be able to update their connectivity lists by removing the clustered nodes.

The cluster formation process is explained with an example. To facilitate the discussion, a simple graph with six SNs and nine edges is used as shown in Figure 4.2. A bidirectional edge from node $i$ to node $j$ indicates that both nodes are directly connected to each other.

![Figure 4.2: A sensor network with six nodes](image)
Algorithm 4.3: CH Selection and Announcement Process

**INPUT:** $C_m$ ; Cluster Node List  
**OUTPUT:** $C_{m-info}$ ; Cluster Head and cluster members information

**STEPS:**

if $\text{round} == 1$ then  
Select the node with maximum energy  
Announce $C_{m-info}$ to the cluster members  
else  
$T_{level} = \frac{\text{avg\_eng}}{2}$; $T_{level}$ indicates threshold value  
if $\text{RSS}(m) < T_{level}$ then  
$\text{max\_eng} = \text{RSS}(m)$;  
$\text{flag} = 0$;  
for each member (i) of $C_m$ do  
if $\text{eng}(i) > \text{max\_eng}$ then  
$\text{New\_CH} = i$;  
$\text{Start\_node} = C_m(1)$;  
$\text{flag} = 1$;  
end  
end  
if $\text{flag} == 1$ then  
interchange between $\text{New\_CH}$ and $\text{Start\_node}$  
anounce $C_{m-info}$ to the cluster members  
end  
end

In this network, when node 1 receives the connectivity information \{1, 2, 3\} from node 5, it ignores the link between node 5 and node 2 since node 1 is not directly connected to node 2. However, it does not decrease the degree of connectivity. Thus, each node forms its connectivity matrix using only common links.

From the connectivity matrix as shown in Table 4.1, each node knows whether it contains the highest degree of connection. Any node containing equal or highest degree of connection can start the cluster formation process after a random time $t$ to avoid collision. For example, in Figure 4.2, both node 1 and node 3 have four connections. Therefore, both of them are eligible to start the cluster formation process. However, only one node will be selected based on the random timer. Suppose, node 1 is selected, then node 1 will inform all its neighbours about its clustering process. The neighbour nodes will wait for a confirmation message from node 1 to
know whether it is selected to join in the cluster formed by node 1.

**Table 4.1: Connectivity matrix of node 1**

<table>
<thead>
<tr>
<th>Node list</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>DoC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

During the waiting period, if any other node attempts to initiate the clustering process, the candidate node will receive a negative acknowledgement from the waiting node. Therefore, the node trying to form a new cluster will also go to wait state until it receives any feedback from the neighbour nodes. After the completion of the clustering process, node 1 will send a message ($C_{info}$: $\{1, 3, 4\}$) to its neighbours including the list of nodes which are nominated as members of that cluster. Each node selected to join in node 1’s cluster will forward the message to all of its neighbours so that they can remove the clustered nodes from their connectivity lists. This ensures that a node, which is already joined in a cluster, cannot be added to another cluster. In this way, every node in the network will be a member node of a cluster. If a node does not have any neighbour node in its connectivity list, then the node itself forms a single node cluster. For example, when node 1 sends cluster information to node 6, node 6 discards node 1 and node 4 from its connectivity list. As a result, it becomes a single node without any link. Hence, node 6 itself forms a single node cluster. Figure 4.3 shows the final output of the cluster formation process. The dotted circle in the figure denotes cluster heads selected on the basis of energy level.

**Effects of defining cluster size:** Since sensor nodes are energy-constrained, the network lifetime is a major concern, specially for applications of WSNs in harsh environments. Equal size clusters can play a significant role in improving the network lifetime since they prevent the exhaustion of the energy of a subset of CHs at a high rate. In the proposed scheme, maximum cluster size can easily be defined by setting
a limit on clique size. Thus, it is possible to divide large networks into smaller clusters and ensure that CHs are not overburdened. Figure 4.4 shows the set of clusters obtained for maximum clique size ‘2’. However, after finishing the cluster formation process, the protocol initiates the setup phase.

**Figure 4.4:** Clusters of maximum clique size [2]

2. **Setup Phase:** Once the CH is selected for a particular cluster and all nodes of that cluster are informed, it is now the responsibility of the CH to assign time slots to all member nodes. In each round, the CH assigns a TDMA slot to the member nodes in the cluster. During the setup phase, every node turns on its own receiver like LEACH. Then, the CH broadcasts an advertisement including the TDMA time slot information. Upon receiving the information, a cluster member goes to sleep state until its time slot starts. Thus, the cluster members transmit the sensed data to the CH only during their time slot. Furthermore, they also send (piggy back) the remaining energy to the CH. Hence, at the end of each round, each CH knows the level of remaining energy of each node in the cluster. After forwarding aggregated
data to the BS, the CH selects the node with maximum remaining energy as the new CH for the next round, if the energy level of the current CH goes beyond the threshold limit (as presented in Algorithm 4.3). The information regarding the new cluster head is broadcast to member nodes at the beginning of the setup phase of the next round.

In the TDMA scheme [144], time is divided into slots and slots are grouped into frames. Let there be $n$ nodes in a cluster, as shown in Figure 4.5, and each node is assigned a slot to transmit sensed data or control messages. In each frame, one free slot is reserved to receive the node joining request from a new node. When a node decides to join a cluster, the node uses this free slot to send its request. At the setup phase of the next round, the CH will broadcast the joining request to all nodes in the cluster. If no negative acknowledgement is received from any cluster member, then the node will be added by the CH and the TDMA frame will be readjusted. In the same way, the CH will periodically check the clique consistency. If any inconsistency is noticed due to mobility or node failure, then the detected node will be removed from the cluster. In the case of failure of the CH, the node with the smallest node ID will restart the CH selection process on the basis of maximum energy. However, once the TDMA schedule is fixed, the routing protocol enters the steady phase.

![TDMA Frame](image)

**Figure 4.5:** TDMA frame within a cluster

### 3. Steady Phase:

In this phase, the CBCR protocol initiates intra-cluster communication. It involves sending sensed data and the energy level to the cluster head during the allocated time slot. The cluster nodes adjust their transmission energy dynamically depending on the received signal strength of the advertisement message. In this phase, if SN has sensed data it will wait for its allocated time slot to forward the data to the CH.
4. **Forwarding Phase**: The inter cluster communication or forwarding phase involves sending aggregated data from the CH to the BS. As CBCR implements multihop routing, in this step, each CH sends data to another CH nearer to the base station. The hierarchical forwarding mechanism described in [48] is adopted in our proposed protocol.

In this scheme, the BS broadcasts a hello message periodically containing two parameters: hop_count (initially set to 0) and energy_level (set to $\infty$ initially). A non CH node just ignores this message. When receiving this message, a CH node adds one to the hop_count and replaces the energy_level with that of the CH, if it is higher than the energy_level of the current CH in the message. In the next step, when another CH receives this message, it will record the previous CH, say ‘x’, and thus ‘x’ will be the forwarding CH to the base station for this CH. If a CH receives another hello message, it will take decisions as per the following rules.

- **Case 1**: hop_count\textsubscript{old} < hop_count\textsubscript{new}, do nothing.
- **Case 2**: hop_count\textsubscript{old} > hop_count\textsubscript{new}, replace the forwarding CH.
- **Case 3**: hop_count\textsubscript{old} = hop_count\textsubscript{new} and energy\textsubscript{old} < energy\textsubscript{new}, replace the forwarding CH, else do nothing.

### 4.2.3 Experiments and Analysis

The CBCR protocol is simulated for two different scenarios. The first scenario is designed to examine the feasibility of the clustering process in terms of average cluster size, coefficient of variance on cluster size, and the number of single node clusters. On the other hand, the second scenario is used to measure the network lifetime and energy consumption in the cluster formation process as well as the routing process. Following are details of the results obtained in the experiments.

**Scenario 1**: In this scenario, the CBCR protocol is simulated in the MATLAB simulator and the results are compared with two leader-first protocols (LCA and
LEACH) as well as with one cluster-first protocol (SDC). LCA is one of the earliest clustering schemes, whereas LEACH is one of the most popular and widely used routing protocols. On the other hand, SDC is a clique based clustering approach like the CBCR protocol. To conduct the experiments, 100, 200, 300, 400 and 500 sensor nodes are randomly deployed in a 100 × 100 (m²) simulation area. The transmission range of all sensor nodes is set to 25 metres except LEACH. The simulation outcome is the average result of 1000 experiments. Furthermore, the effect of defined clique size in the CBCR protocol in the cluster formation process is also investigated and presented. The limit on cluster size is set on the basis of the assumption that no cluster will contain more than 10% of the total nodes in the network. This strategy protects the cluster heads from being overburdened and thus improves network stability. The following performance metrics are used to evaluate cluster characteristics in the experiments: average cluster size, coefficient variance of cluster size, number of single node clusters, and maximum cluster size.

Figure 4.6 presents the average cluster size of the evaluated protocols. In sensor networks, average cluster size is preferred not to be too small in order to ensure optimal clustering. On the other hand, average cluster size should not include too many nodes in a cluster to avoid message collisions and transmission delay. However, it can be seen that the average cluster size increases with the increase in node density in the network for each protocol. The increment is almost linear and the amount is significant in both LEACH and LCA, while the increment is comparatively lower in the case of CBCR and SDC protocols. The reason is that all nodes in a cluster in the CBCR protocol are located at 1-hop distance, whereas the maximum distance between any two nodes in one cluster is two hops in LCA. As a result, CBCR has a smaller average cluster size compared to the LCA protocol. Like LCA, the same trend is observed for the LEACH protocol. This is due to the probabilistic cluster head selection procedure. If the selected cluster heads are located very near to each other, there will definitely be some clusters with a large number of nodes. On the other hand, SDC has the lowest average cluster size, and thus the protocol incurs communication overheads.
Figure 4.6: Average cluster size vs number of nodes

Figure 4.7 presents the variability of the cluster sizes formed by the routing protocols. The Coefficient of Variance is measured as follows: $100 \times \frac{\text{Standard deviation}}{\text{Mean value of set}}$. The result shows that CBCR has a smaller coefficient variance than the other three protocols. This indicates that the proposed protocol forms uniform clusters. In LCA, the CHs are selected arbitrarily, and they form clusters by including all non-clustered SNs within their communication range. On the other hand, LEACH implements a probabilistic approach to determine the CHs and the nodes choose the minimum distance CHs to forward the data packets. Thus, these approaches cannot maintain a balance in the number of SNs in different clusters. The same results are obtained for the SDC protocol since it implements the global maximal clique formation process. However, the CBCR protocol forms clusters locally using topology information, and thus it ensures a low variance in cluster sizes.

Figure 4.8 shows the number of single node clusters generated by the above four protocols. A good clustering approach must maintain a minimum number of single node clusters to provide effective and cooperative communication among
Chapter 4. *Energy Efficient Single Path Routing*

Figure 4.7: Coefficient of variance on cluster size

Figure 4.8: Number of single node clusters
the sensor nodes in a network. It can be seen that CBCR has fewer single node clusters compared to LEACH, LCA, and SDC protocols. LCA generates the largest cluster first, and thus the protocol results in many single node clusters. Similarly, in LEACH, if boundary nodes or nodes in the same region are selected as CHs, the number of single node clusters increases. In SDC, global maximum cliques are obtained to form the clusters, which leads to a significant number of single nodes in the network. For densely deployed sensor nodes in a network, CBCR has a minimum number of single node clusters according to the simulation result.

Finally, we examined the effect of undefined and defined clique size for the proposed protocol. The result in Figure 4.9 indicates that a cluster can contain more than sixty nodes, if the clique size is not defined. In such a situation, the cluster head has to perform a significant amount of in-network processing, and thus will quickly deplete its energy. The CBCR protocol can easily avoid this problem by defining the clique size as it is not possible to form a cluster greater than the pre-defined size.

![Figure 4.9: Maximum cluster size](image-url)
**Scenario 2:** In this scenario, energy efficiency and network lifetime of the CBCR protocol is compared with two routing protocols: LEACH, and EEGA. LCA and SDC protocols mainly focus on clustering and therefore, are not considered in performance evaluation of routing protocols. To compute the energy dissipation in the clustering process, 100, 200, 300, 400, and 500 SNs are randomly deployed in the network area, where non CHs continuously send data during their assigned time slots. The other experiments are done with 100 nodes to keep the tests simple. Furthermore, the experiments are run for 1000 simulation trials to compute the average value.

Table 4.2 lists the simulation parameters used in the experiments. The base station is located at the centre (50, 50) of a $100 \times 100 \, m^2$ area, and it has infinite power supplies. On the other hand, the initial energy of each sensor is set to 0.5 joules. For application layer parameters, the transmission range is set to 20 metres for both the CBCR and EEGA protocols, whereas SNs in LEACH are able to directly communicate with the CHs. Furthermore, the size of control packets and the packet header of data packets is assumed to be 24 bytes long, whereas the payload size of data packets is set to 128 bytes. Finally, the radio parameters for free space and multipath propagation are assigned the same values as defined in the Heinzleman et

<table>
<thead>
<tr>
<th>Types</th>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>Network Grid</td>
<td>(0,0) to (100, 100)</td>
</tr>
<tr>
<td></td>
<td>Initial Energy</td>
<td>0.5J</td>
</tr>
<tr>
<td></td>
<td>Location of BS</td>
<td>(50, 50)</td>
</tr>
<tr>
<td>Application</td>
<td>Cluster Radius</td>
<td>20m (except LEACH)</td>
</tr>
<tr>
<td></td>
<td>Data Packet Size</td>
<td>128 bytes</td>
</tr>
<tr>
<td></td>
<td>Broadcast Packet Size</td>
<td>24 bytes</td>
</tr>
<tr>
<td></td>
<td>Packet Header Size</td>
<td>24 bytes</td>
</tr>
<tr>
<td>Radio Model</td>
<td>$E_{elec}$</td>
<td>50 nJ/bit</td>
</tr>
<tr>
<td></td>
<td>$E_{fs}$</td>
<td>10 pJ/bit/m²</td>
</tr>
<tr>
<td></td>
<td>$E_{mp}$</td>
<td>0.0013 pJ/bit/m⁴</td>
</tr>
<tr>
<td></td>
<td>EDA</td>
<td>5 nJ/bit/signal</td>
</tr>
<tr>
<td></td>
<td>Threshold Distance ($d_0$)</td>
<td>87.71m</td>
</tr>
<tr>
<td>Duration</td>
<td>Round</td>
<td>Time 15s</td>
</tr>
</tbody>
</table>
al. model [43], and the duration of each round is set to 15 seconds.

Figures 4.10-4.13 show the total energy dissipation in the cluster formation process for the LEACH, EEGA, and CBCR protocols. Figure 4.10 represents the initial energy consumption of these three protocols, whereas Figure 4.11, Figure 4.12, and Figure 4.13 illustrate the amount of energy dissipation after 25, 50, and 75 rounds respectively. These experiments are performed to evaluate the performance of clustering process for different numbers of sensor nodes.

It can be seen that CBCR and EEGA consume more energy than LEACH initially. This is because both protocols exchange topology information to all neighbour nodes at the beginning. In contrast, only CHs send the control messages in LEACH, which dissipates very low energy in the cluster formation process. However, the energy consumption in the CBCR protocol decreases after the initial round. The reason is that CBCR then only rotates the CHs in each round on the basis of the threshold limit and maximum energy. Hence, CBCR protocol has to send very few messages in each round. On the other hand, LEACH restarts the clustering process at the beginning of each round. Therefore, energy consumption in LEACH exceeds CBCR after a few rounds. Similarly, EEGA consumes more energy than CBCR in each scenario since every node in EEGA forms its own clique and then exchanges the information to form global cliques. As a consequence, EEGA sends and receives more control messages compared to the CBCR protocol.

Figure 4.12 and Figure 4.13 show that the LEACH protocol consumes about 3J and 4.5J of energy after 50 and 75 rounds respectively for 500 nodes, whereas the amount is 1.8J and 2J respectively for CBCR. This indicates that there is about 50% increase in energy use by the LEACH protocol after every 25 rounds. The EEGA protocol also dissipates slightly more energy compared to the CBCR protocol due to execution of redundant clique formation process. Thus, CBCR outperforms the other two protocols in terms of energy dissipation in the clustering process. Furthermore, the figure shows that the amount of total energy dissipation increases with the increase of node density in the network. Since the network area is fixed,
Figure 4.10: Initial energy dissipation

Figure 4.11: Energy dissipation after 25 rounds
Figure 4.12: Energy dissipation after 50 rounds

Figure 4.13: Energy dissipation after 75 rounds
an increase in the number of sensor nodes results in the exchange of more control packets among the neighbour nodes.

The total number of nodes which are exhausted of their energy over a simulation time of 2000 trials is depicted in Figure 4.14. The figure shows that nodes remain alive for a longer time in CBCR compared to LEACH and EEGA protocols. It can be seen that LEACH becomes non-functional after about 1500 rounds whereas the CBCR protocol remains operative for more than 2100 rounds. The reason for the longer time is that CBCR conserves a significant amount of energy by reducing unnecessary clique formation and CH selection processes. Furthermore, the protocol selects maximum energy cluster heads and ensures a balance in energy consumption in the network. In LEACH, low energy nodes can also be selected as CHs and thus they quickly exhaust their energy. Like LEACH, the EEGA protocol also selects low energy nodes as CHs. Moreover, the protocol exchanges many control packets to form the global maximum clique for every SN. Therefore, nodes in CBCR are alive for more rounds with respect to the LEACH and EEGA protocols.

Figure 4.14: Comparison of network lifetime for LEACH, CBCR, and EEGA
Figure 4.15 presents the total energy consumption by the LEACH, EEGA, and CBCR protocols. It can be seen that LEACH and EEGA consume more energy than CBCR due to re-clustering and transmitting excessive control messages at each round. On the other hand, although the energy use in the CBCR protocol is high initially, the amount decreases with the increase of rounds. Optimisation of the clique formation process and elimination of unnecessary clustering with the help of threshold limit at each round improves the energy conservation in the CBCR protocol. Furthermore, LEACH and EEGA protocols waste a significant amount of energy since the network loses its connectivity after about 1500 and 1900 rounds respectively. Thus, the CBCR protocol better utilises node energy compared to other two protocols.

![Figure 4.15: Total energy dissipation for LEACH, CBCR, and EEGA](image)

Finally, Figure 4.16 shows a comparison of the network lifetime using three different metrics: i) First Node Dies (FND), ii) Half of the Nodes Dies (HND), and iii) 90% of the Node dies, that is, Most of the Nodes Dies (MND) [145]. The result indicates that the proposed protocol can increase the lifetime of a sensor network.
by 64% for FND, 27% for HND, and 40% for MND over LEACH; and 8% for FND, 3% for HND, and 4% for MND over the EEGA protocol.

![Network lifetime with respect to FND, HND, and MND](image)

**Figure 4.16:** Network lifetime with respect to FND, HND, and MND

### 4.2.4 Discussion

The CBCR protocol is an energy efficient protocol which first discovers the neighbour nodes of a SN and then starts the clustering process on the basis of maximum degree of connectivity. Instead of selecting new cluster heads at every round, CBCR determines maximum energy cluster heads depending on a pre-defined threshold value. Thus, the protocol reduces the number of clique formation and cluster head selection processes, which leads to a significant amount of energy conservation. The CBCR protocol can widely be used in data-driven applications (e.g., data fusion, passive monitoring) for large-scale sensor networks.
4.3 Interference Aware Heuristic Routing Scheme

Wireless sensor networks are mainly data centric and application specific. Hence, the design requirements of a sensor network vary from application to application. For example, the requirements of a passive monitoring application include fewer constraints compared to an energy conserving application. Therefore, the general routing protocols designed for large-scale sensor networks may not cope with the challenges which have significant impacts on small-scale networks. This section presents an interference aware heuristic routing protocol developed for small-scale networks such as wireless home automation networks.

4.3.1 Wireless Home Automation Networks

Among the wide variety of WSN applications, WHANs have gained considerable attention both in academia and industry over the past few years. Moreover, recent research indicates that it will be one of the fastest growth sectors in consumer electronics over the next five years [15]. A WHAN consists of a number of sensors and actuators which sense and forward collected information to the base station in order to monitor and control devices in a home network. As depicted in Figure 4.17, the WHAN contains a gateway server, a base station, and a number of sensor nodes. The SNs are spread over the house to collect various data and send it to the BS via other sensors within one hop distance. The BS is responsible for the maintenance of the entire network, whereas the gateway server connects the BS with the end user in order to provide external controls. Compared to the battery operated SNs, the BS has more computation, memory, and energy resources. Although the battery powered SNs are easy to install, they need to be replaced or recharged on a regular basis. Statistics show that in about 50% of in-home displays (IHD), the batteries are not replaced when they no longer have any power left [146]. Therefore, an energy efficient routing protocol is essential to improve the lifetime of WHANs.

A number of standard solutions have been proposed for WHANs such as ZigBee, Z-Wave, INSTEON, Wavenis, Bluetooth, WiFi, and UWB (Ultra-Wideband) [147,
Figure 4.17: An example of wireless home automation networks

106. However, these protocols have a number of limitations. First, very few of them consider link quality as a routing metric. For example, Z-Wave, INSTEON, Bluetooth, and UWB are susceptible to interference and is not suitable for the applications communicating over noisy channel. Similarly, Wavenis uses Received Signal Strength Indicator (RSSI) value to estimate link quality which may be inaccurate because of multipath and interference [147]. Second, most of these WHAN architectures (ZigBee, INSTEON, and UWB) implement flooding-based routing protocols that are subject to high energy consumption and high probability of collision in the MAC layer. Third, these protocols reduce the total energy consumption by sacrificing uniform energy use in the network and thus result in quick network partitions. In addition, WHAN is subject to interference due to the presence of numerous devices and networks in home environments such as WiFi routers, smart meters, cordless phones, and microwave ovens [149]. To ensure end-to-end connectivity, multihop communications are required to retransmit data for nodes that are not within the
sender’s transmission range [150]. Although interference is a common problem in home environment, very few protocols consider interference as one of the important parameters in a WHAN routing protocol design, in contrast to routing protocols in wireless sensor networks.

To overcome these limitations, we present IAHR, a heuristic routing protocol based on both local and global knowledge, which significantly improves the network performance. Furthermore, instead of considering interference as a binary event (i.e., yes or no), the IAHR protocol quantises the level of interference in each link and then uses the value as one of the routing parameters. Thus, the protocol takes every opportunity to deliver data packets using low-interfered links and keeps the network functional. Finally, we investigate the effects of random changes in topology and demonstrate the performance of the proposed protocol with the standard specification.

4.3.2 Network Model

It is assumed that a WHAN is comprised of a number of SNs, a BS, and a GS. Further, the following assumptions are considered to define the network model.

- Every SN has a unique node ID and the SN is aware of the location of the BS as well as its own.

- SNs periodically send remaining energy and link quality (LQ) values to all of their neighbour nodes.

- SNs are equipped with limited and non-chargeable batteries, whereas the BS has a sufficiently large amount of power.

- The network consists of both stationary and mobile nodes and the energy consumption is not uniform for every SN.
Chapter 4. Energy Efficient Single Path Routing

4.3.3 Radio Model

The energy consumption model for radio communication is assumed to be similar to the Heinzelman et al. radio model [43]. We have used the multipath propagation model based on the distance between the transmitter and the receiver. In order to receive a \( k \)-bit message and transmit a \( k \)-bit message over the distance \( d \), the radio hardware dissipates energy as per the following equations respectively.

\[
E_{Rx}(k) = k \epsilon_{elec} \tag{4.1}
\]

\[
E_{Tx}(k, d) = k \epsilon_{elec} + k \epsilon_{amp} d^\gamma \tag{4.2}
\]

where, \( \epsilon_{elec} \) and \( \epsilon_{amp} \) denote electronics energy and amplifier energy respectively, and \( \gamma \) represents the path-loss exponent. The value of \( \gamma \) is set to 3, which is typical for home and office environments [151].

4.3.4 The IAHR Protocol

A WHAN may have both static and mobile nodes. It is assumed that the location of the static nodes is determined during the network deployment phase using either physical devices (e.g., Global Positioning System) or topology discovery algorithms [152]. Every SN exchanges the longevity factor (LF) and the location information with its neighbour nodes during the network setup phase, whereas the location of the BS is broadcast to all SNs. The network is considered to be interference free at this stage, and therefore, the \( LQ \) value is set to zero for all SNs initially. In order to adapt to changes in the network topology, the SNs share the \( LF \) and \( LQ \) value at the beginning of every round with their neighbour nodes. Since WHAN is a small network, the overhead of exchanging local information is reasonable.

Node mobility is another critical issue in WHANs. It is likely that the number of mobile nodes is smaller compared to the static nodes in a home network. In the IAHR protocol, the static nodes keep separate records of the mobile nodes and they usually do not select mobile nodes to route data packets. When a mobile node has
to transmit data packets, it sends a ‘Hello’ message to discover the neighbour nodes within its communication range. Upon receiving the reply messages from neighbour nodes with location information, the mobile node selects the nearest neighbour to forward its data packets. However, the static nodes may also use the mobile nodes to route data packets if, and only if, there is no route available to reach the BS through the static sensors. Thus, the IAHR protocol makes every effort to keep the network functional.

4.3.4.1 Heuristic Function

The heuristic function guides the search procedure by providing a good estimate of how far the destination node is from the current node. It derives a numerical value for every SN by combining different routing parameters. The following parameters are used in the heuristic function of the IAHR protocol.

a) Local and Global Distance- The local and global information on network topology helps to find an efficient route. Since the use of only local or global information may lead to network partition and local minimum problems, the IAHR protocol uses the distance traversed so far (i.e., local distance), as well as the estimated distance between a SN and the BS (i.e., global distance), to guide the search procedure in the right direction.

b) Longevity Factor- Remaining energy is an important routing metric which can help to extend network lifetime by avoiding low energy nodes during the route selection process. The IAHR protocol derives a longevity factor for every SN from the remaining energy using the following equation:

\[
LF = \frac{T \times E_{res}}{E_{init}}
\]

(4.3)

where, \( T \) is the sustainable time of a SN, \( E_{res} \) and \( E_{init} \) are remaining energy and initial energy of that SN respectively. The \( LF \) is used to maximise the value of the heuristic function to avoid low energy nodes from being selected repeatedly.
Figure 4.18 presents a network topology with nine SNs and a BS. Let the SNs send information to the BS using a greedy forwarding algorithm proposed in the GAHR protocol. The greedy algorithm uses the global distance between SNs and the BS to select the forwarding nodes. Suppose, the global distance between node 4 and the BS is 25, whereas the distance is 22 for node 5. Hence, all packets from node 1-3, 8 and 9 will go through node 5 in this method. As a result, node 5 will deplete its energy and node 8 and node 9 will be disconnected from the network. Even the use of the A* search algorithm fails to overcome this situation if the route ‘node 5 – node 6 – BS’ is better than the route ‘node 4 – node 7 – BS’. However, the proposed protocol overcomes this problem by using the longevity factor in addition to the local and global distance.

c) LQ Metric- Link quality in WHANs is highly affected by interference. Interference leads to packet retransmissions and thus wastes a significant amount of energy. However, the existing protocols always consider interference as a binary event, although it has a continuous degree of impact on the reception of packets based on the Signal-to-Interference-and-Noise Ratio (SINR) value [153].

Figure 4.18: Impacts of the greedy forwarding technique
Most of the network models assume that a packet is received correctly if the \( S\text{INR} \) value is above the threshold level. However, the research indicates that the \( S\text{INR} \) threshold is not a constant value, and it mainly depends on the transmitter hardware and the signal strength level [154]. For example, the experimental results on the Mica2 sensor mote find the following \( S\text{INR} \) thresholds for successful packet receptions: \( \text{SINR}_{\text{LOW}} = 2\text{dB} \); \( \text{SINR}_{\text{HIGH}} = 6\text{dB} \). The \( \text{SINR}_{\text{LOW}} \) threshold level indicates that if the \( S\text{INR} \) is equal to or less than 2dB then the packet cannot be decoded. On the other hand, if the \( S\text{INR} \) value is equal to or greater than 6dB, the packet will be received successfully. It is noted that the probability of successful packet reception increases with the increase of the \( S\text{INR} \) value. As an example, if the \( S\text{INR} \) value is 5dB, there is more than 80% probability that a packet will be decoded correctly, whereas the probability is about 10% for 4dB \( S\text{INR} \) [154]. The IAHR protocol takes advantage of this \( S\text{INR} \) characteristic and quantifies the \( S\text{INR} \) level in such a way that the value can directly be used in the route selection process.

In IAHR, the receiver node calculates the level of interference using the geometric \( S\text{INR} \) model [155] at each round. For example, if \emph{node} \( v \) sends data packets to \emph{node} \( u \), then the \emph{node} \( u \) measures the \( S\text{INR} \) of link ‘\( u-v \)’ as follows:

\[
S\text{INR}(uv) = \frac{S_{vu}}{I_{vu} + N}
\]  

(4.4)

where, \( S_{vu} \) denotes the signal strength of sender \( v \), \( I_{vu} \) is the total signal power received from other nodes except \( v \), and \( N \) is the noise power. The value of \( S_{vu} \) is the product of the path gain (\( G_{vu} \)) and the power (\( P_v \)), whereas \( I_{vu} \) is the sum of the signal power of other nodes, i.e., \( I_{vu} = \Sigma G_{wu}P_w \), for \( w \neq v \). On the basis of the \( S\text{INR} \) value, each node is assigned a numerical value which presents the quality of the link between the sender and the receiver. For example, if \( S\text{INR} \geq 6\text{dB} \), the link is not affected at all. In contrast, the link is completely down when \( S\text{INR} \leq 2\text{dB} \). Thus, the effect of interference and noise on a link (\( u-v \)) is defined as per the
Chapter 4. *Energy Efficient Single Path Routing*

The following equation:

\[
LQ(u-v) = \begin{cases} 
0 & ; \text{SINR}(uv) \geq 6 \\
100 & ; \text{SINR}(uv) \leq 2 \\
(SINR_{\text{HIGH}} - SINR_{u-v})^2 & ; \text{otherwise}
\end{cases}
\] (4.5)

The \( LQ \) value is set to 100 if the \( \text{SINR} \) is equal to or less than the lower threshold. This is done to increase the heuristic value of the interfered link so that the link becomes unavailable for the current round. Similarly, the closer the \( \text{SINR} \) value is to the upper threshold, the lower the \( LQ \) value is used to represent it. As an example, if the \( \text{SINR} \) of a link is 5dB, then the \( LQ \) value will be \((6 - 5)^2\), i.e., 1. Thus, the IAHR protocol represents the interference using different weights on the basis of the link quality. Now, combining the heuristic function parameters, node \( v \) computes the heuristic value of node \( u \) as follows:

\[
f_v(u) = g(u) + h(u) + \frac{100}{LF(u)} + LQ(u-v)
\] (4.6)

where, \( f_v(u) \) is the heuristic value of node \( u \), \( g(u) \) and \( h(u) \) are local and global distance respectively, \( LF(u) \) is the longevity factor and \( LQ(u-v) \) is the quality of the link between node \( u \) and node \( v \). The inverse of the \( LF \) is multiplied by 100 to increase the heuristic value of low energy nodes in the equation.

### 4.3.4.2 The \( A^* \) Search Algorithm

The \( A^* \) search combines the best-first-search and Dijkstra’s algorithm to find an efficient route between source and destination nodes. This is a leading path-finding algorithm used in real-world applications such as the Warcraft III game, and road networks [156]. The \( A^* \) search is more efficient than other conventional shortest path algorithms such as Bellman-Ford and Dijkstra’s algorithms. The conventional search algorithms are not applicable in WHANs due to long delays and computational complexity. It is difficult for them to adapt to frequent changes in the network topology. Furthermore, the conventional algorithms may be trapped into a blind
search and therefore, consuming a significant amount of resources. In contrast, the
$A^*$ search algorithm explores a number of choices so that it can backtrack to a earlier
state to overcome such situation. Thus, the algorithm can effectively deal with the
local minimum problem and direct the search procedure to the right direction. To
efficiently compute the optimal solution, it combines features of both pure heuristic
search and uniform cost search.

Algorithm 4.4: $A^*$ search algorithm

INPUT: start_node, goal_node
OUTPUT: path_list

STEPS:

open_list $= \{ \text{start}_\text{node} \}$; list of explored nodes
closed_list $= \{ \emptyset \}$; list of visited nodes, initially empty
path_list $= \{ \emptyset \}$; list of selected nodes, initially empty

while $\text{open}_\text{list} \neq \emptyset$ do
    select the node with lowest $f$-value in open_list
    make it current_node and remove from the open_list
    if current_node is goal_node then
        return path_list;
    else
        add the current_node to the closed_list
        for each neighbour of current_node do
            if the neighbour is in the closed_list then
                continue
            end
            compute the sum of local distance, LF and LQ
            if the neighbour is in the open_list & new_sum < old_sum then
                update the parent_info in the path_list
                update the sum and $f$-value of the neighbour
            end
            if the neighbour is not in the open_list or closed_list then
                add the neighbour in the open_list
                add the parent_info in the path_list
            end
        end
    end
end

The $A^*$ search algorithm maintains an open_list and a closed_list as shown in Al-
gorithm 4.4, where the open_list initially contains the start_node and the closed_list
is empty. Then it removes the first node from the open_list and makes it the
current_node. After that, the algorithm explores all neighbours of the current_node, computes their heuristic values, and adds them to the open_list. The open_list is then sorted in ascending order with respect to the heuristic value, and then the current_node is added to the closed_list as well as in the path_list. The path_list is an array which keeps track of the valid path by updating the visited nodes accordingly. Now, it again removes the first node from the open_list, makes it the current_node, and follows the same procedure until the goal state is found or the open_list is empty.

In a nutshell, the A* search follows a path of the lowest known heuristic cost and also maintains a priority queue of alternative path segments along the way. Thus, it is not trapped in an infinite loop and reduces delay and complexity. Since WHAN consists of resource-constrained SNs, the A* heuristic search algorithm is more suitable compared to the conventional algorithms.

### 4.3.4.3 Routing Procedure

The heuristic function of the proposed IAHR protocol uses four parameters to calculate the heuristic value $f$ of a node: the distance traversed to reach the current node from the source node ($g$), the Euclidian distance between the current node and the BS ($h$), the longevity factor ($LF$), and the link quality ($LQ$). As an example, Figure 4.19 depicts a network topology with 15 SNs and a BS. The communication range of the SNs is presented by the dotted circles, whereas the dotted lines represent the communication links between two nodes. Every node computes the route to reach the BS using the A* search algorithm. In order to understand the impacts of remaining energy and interference in WHANs, three different cases are explained for the proposed protocol.

The first case assumes that all SNs have the same level of energy and there is no interference in the network. Suppose, node 6 has to find the route to reach the BS and the initial parameters of its neighbour nodes are as follows: $g(5) = 1, g(7) = 3$; $h(5) = 10, h(7) = 11$; $eng(5) = eng(7) = 0.5 \text{ J}$; $LQ(5-6) = LQ(7-6) = 0$; $T = 50 \text{ minutes}$ for both nodes. First, node 6 computes its $f$-value via node 5 and node 7 as follows: $f(5) = 1 + 10 + 2 + 0 = 13, f(7) = 3 + 11 + 2 + 0 = 16$. 


Since the $f$-value via node 5 is smaller than that of node 7, node 6 selects node 5 as its forwarding node. Upon receiving the information, all neighbour nodes update their $f$-values with respect to node 6. During the next hop selection process, node 5 calculates the $f$-value to reach the BS via node 7 and node 4. Let the values be: $f(4) = 11; f(7) = 15$ respectively. Hence, node 4 is selected as the next hop as well as the $f$-value and $g$-value of node 7 are also updated. In this way, the route of node 6 is constructed to reach the BS.

The second case assumes that the SNs have different level of energy and the network is still interference free. Suppose, after time $t$, the energy level of node 5 and node 7 degrades to 0.1 J and 0.25 J respectively. At this stage, the new $f$-values of node 5 and node 7 are: $f(5) = 1 + 10 + 100/((0.1 \cdot 50)/0.5) + 0 = 21$, and $f(7)$
\[= 3 + 11 + \frac{100}{((0.25 \cdot 50)/0.5)} + 0 = 18\] respectively. Since the heuristic value of node 7 is smaller than that of node 5, node 7 will be selected to forward the data packets of node 6. It means that the IAHR protocol avoids low energy nodes when there exists alternative routes. Thus, the protocol ensures energy balanced routing in WHANs.

The third case explains the effects of both remaining energy and interference. Consider, the following parameters of node 5 and node 7: \([\text{eng}(5) = 0.2, \text{eng}(7) = 0.5]; [LQ(5-6) = 0, LQ(7-6) = 4]\). Using the values, node 6 computes the heuristic values of node 5 and node 7 as follows: \(f(5) = 1 + 10 + \frac{100}{((0.2 \cdot 50)/0.5)} + 0 = 16\), and \(f(7) = 3 + 11 + \frac{100}{((0.5 \cdot 50)/0.5)} + 4 = 20\). Although node 7 possesses more energy than node 5, it will not be allowed to forward data packets. It indicates that the IAHR protocol effectively handles the interference problem and minimises packet retransmissions.

4.3.5 Experiments and Analysis

The proposed protocol is implemented in the OMNeT++ simulator and compared with the AODV\textsubscript{JR} and GAHR protocols. AODV\textsubscript{JR} is a widely used standard for WHANs, whereas the GAHR protocol incorporates interference in addition to node mobility, local and global distance. Five performance metrics are chosen to evaluate and compare the efficiency and reliability of the IAHR, GAHR, and AODV\textsubscript{JR} protocols. They are Packet Delivery Ratio (PDR), Remaining Energy, Network Lifetime, Routing Overhead, and Average Packet Delay. The simulation parameters used in the experiments are listed in Table 4.3.

Two different scenarios are designed to evaluate the performance of the above three protocols. The number of mobile nodes in both scenarios is two, and the number of interferer nodes is two in the first scenario, and three in the second scenario. To observe the impacts of increased network connectivity, the experiments are performed for 50, 60, and 70 nodes respectively. Furthermore, the simulation is performed 500 times for every experiment to compute the average value.
Table 4.3: Simulation parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Grid</td>
<td>(0, 0) to (100, 100)</td>
</tr>
<tr>
<td>Location of BS</td>
<td>(50, 50)</td>
</tr>
<tr>
<td>Initial Energy</td>
<td>2 J</td>
</tr>
<tr>
<td>Transmission Range</td>
<td>20 m</td>
</tr>
<tr>
<td>Data Packet Size</td>
<td>128 bytes</td>
</tr>
<tr>
<td>Broadcast Packet Size</td>
<td>24 bytes</td>
</tr>
<tr>
<td>$\epsilon_{\text{elec}}$</td>
<td>50 nJ/bit</td>
</tr>
<tr>
<td>$\epsilon_{\text{amp}}$</td>
<td>10 pJ/bit/m$^2$</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>3 (lossy medium)</td>
</tr>
</tbody>
</table>

4.3.5.1 Performance Comparison

The Scenario 1 is designed to investigate the PDR, network lifetime, and total energy consumption in the IAHR, GAHR, and AODV$_{jr}$ protocols. This scenario uses two mobile nodes and two interferer nodes. The speed of both mobile nodes is set to 0.5 metre/second. One interferer node generates random levels of interference by varying its transmission power from -10dB to 2dB, whereas the transmission power of the other node is fixed to -10dB.

Figure 4.20 illustrates the packet delivery ratios in the IAHR, GAHR, and AODV$_{jr}$ protocols. It can be seen that the IAHR protocol achieves a higher level of PDR than the other two protocols. The GAHR protocol mainly uses the greedy search algorithm to construct the routes. Since greedy forwarding results in quick network partitions, many SNs are disconnected and unable to reach the BS. This does not happen in the IAHR protocol since it maintains a balance in energy consumption of the SNs. The heuristic function of the IAHR protocol maximises the heuristic values of energy critical nodes, and thus finds alternative routes to send the data packets. On the other hand, the AODV$_{jr}$ protocol floods the route request packets on the entire network, which leads to heavy contention and collisions in the MAC layer. The sensors in AODV$_{jr}$ deplete their energy quickly and partition the network. Furthermore, the results also indicate that the PDR is increased with an increase in the number of sensors in each protocol. Since the network area is fixed, the growth
in node density increases the network connectivity as well as the total number of delivered packets.

Figure 4.21 shows the amount of remaining energy in the IAHR, GAHR, and AODV$_{jr}$ protocols until the network becomes non-functional. The network is considered non-functional when 10% of the total nodes run out of power. It can be seen that the amount of remaining energy is highest in the GAHR protocol. This remaining energy is wasted since the network is down and unable to continue its operation at that time. The GAHR protocol does not consider node energy to compute the route. Hence, the protocol may use the critical nodes to forward data packets until they exhaust their energy. The graph also shows that the amount of remaining energy in AODV$_{jr}$ is the lowest among them all. However, this result does not indicate the energy efficiency of the AODV$_{jr}$ protocol. The reason is that the SNs in AODV$_{jr}$ waste a significant amount of energy due to the implementation of the flood-based route discovery mechanism. The frequent packet collisions and retransmissions are mainly responsible for energy drainage in AODV$_{jr}$. Since IAHR implements energy...
balanced routing, the nodes remain alive for more rounds and they consume more energy due to the increased volume of data transmissions compared to the GAHR and AODV\textsubscript{jr} protocols.

The simulation result on the network lifetime comparison among the IAHR, GAHR, and AODV\textsubscript{jr} protocols is illustrated in Figure 4.22. It can be seen that the network (with 50 nodes) remains functional for 713 rounds in the IAHR protocol, whereas the number is 634 and 119 for the GAHR and AODV\textsubscript{jr} protocols respectively. This similar trend is observed for the networks consisting of 60 and 70 nodes respectively. Furthermore, the graph shows that the network lifetime decreases with an increase in the number of nodes in the network. This increased number of nodes results in an increased volume of data traffic, which leads to a high probability of collisions in the network. Even in this situation, the IAHR protocol outperforms the other two protocols. Every node in the IAHR protocol is aware of the link quality and remaining energy of its 1-hop neighbours at the beginning of each round. This local information helps the protocol to avoid unsuccessful data
transmission through high-interference links. Furthermore, the exclusion of low energy nodes in the route selection process delays the network partition. Therefore, nodes remain alive for more rounds in IAHR than the other two protocols.

4.3.5.2 Performance Comparison: High Interference Scenario

In Scenario 2, the experiments are performed to measure the PDR, routing overhead and average packet delay. The AODV$_{jr}$ protocol is not evaluated since the protocol does not consider the link quality in the route selection process. However, this scenario uses three interferer nodes where two nodes have the same properties as defined in Scenario 1. The third interferer node is mobile and generates random levels of interference.

Figure 4.23 indicates that the IAHR protocol achieves 10-15% more PDR gain compared to the GAHR protocol in a noisy communication environment. In IAHR, every SN is aware of the link quality of its one hop neighbours and can easily select alternative nodes to avoid the interfered links. If the interfered links are recovered, then the information is provided to the neighbour nodes at the beginning of the next round. Therefore, the links can again be used in forwarding data packets.
On the other hand, the GAHR protocol marks a link as permanently unstable if it exceeds the threshold level of the recovery count. This approach reduces the node connectivity, and thus many packets are unable to reach the BS.

Table 4.4 represents the routing overhead and average packet delay of the IAHR and GAHR protocols. It can be seen that the GAHR protocol sends more control packets compared to the IAHR protocol. The GAHR protocol follows the greedy forwarding technique and restarts the route discovery process using the \( A^* \) search algorithm to overcome the local minimum problem. Furthermore, since the protocol detects interference in real time, it has to find new routes to avoid the interfered links. Therefore, the GAHR protocol has to send more control packets than the IAHR protocol. For the same reason, the average packet delay in GAHR is also higher than that of the IAHR protocol. Instead of marking a node *unstable*, the IAHR protocol quantises the level of interference in order to calculate the minimum interfered links. In the presence of mobile interferers, this strategy significantly minimises the delay since the affected links become functional as soon as they recover from interference.
Table 4.4: Comparison on routing overhead and latency

<table>
<thead>
<tr>
<th>Protocols</th>
<th>Total Nodes</th>
<th>Routing Overhead</th>
<th>Avg. Packet Delay (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAHR</td>
<td>50</td>
<td>1611</td>
<td>0.0207</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>1783</td>
<td>0.0236</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>2276</td>
<td>0.0254</td>
</tr>
<tr>
<td>GAHR</td>
<td>50</td>
<td>1634</td>
<td>0.0317</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>1857</td>
<td>0.0343</td>
</tr>
<tr>
<td></td>
<td>70</td>
<td>2388</td>
<td>0.0392</td>
</tr>
</tbody>
</table>

4.3.6 Discussion

The IAHR protocol uses the $A^*$ search algorithm to find an efficient route on the basis of a number of routing parameters. In IAHR, every SN is aware of the local and global distance, remaining energy, and link quality of its neighbour nodes before starting the route selection process. Therefore, low energy nodes and high-interference links can be avoided when selecting the routes. Thus, the network lifetime is improved over that of the GAHR and AODV$_{jr}$ protocols by a factor up to 1.84 and 6.36 respectively. Similarly, the PDR of the IAHR protocol is increased by a maximum of 10% and 25% over the GAHR and AODV$_{jr}$ protocols respectively. Although the IAHR protocol is designed for WHANs, it can also be used in large-scale WSN applications because of its self-adaptation characteristic.

4.4 Summary

Hierarchical and location-based single path routing protocols are more energy efficient compared to the flat-based single path routing schemes. Therefore, this study has proposed two routing schemes: i) hierarchical clique-based protocol, and ii) location-based interference aware protocol. Most of the clique-based protocols waste a significant amount of energy by initiating unnecessary clique formation and clustering processes at each round. However, the CBCR protocol shows that the
maximal clique formation process on the basis of the degree of connectivity results in lower number of single node clusters. Furthermore, this process is more energy efficient compared to the global clique formation process. On the other hand, the IAHR protocol considers interference as one of the important routing parameters and quantises the level of interference instead of considering it as a binary event. Thus, the protocol makes every effort to deliver data packets using low-interfered links and keeps the network functional. This strategy significantly improves the network lifetime and PDR in noisy communication environments. Experimental results show that the CBCR protocol improves the network lifetime by an amount of 4% and 40% over the EEGA and LEACH protocols respectively. Similarly, the PDR of the IAHR protocol improves by an amount of 10% over the GAHR protocol and 25% over the AODV$_{jr}$ protocol.

One limitation is that the proposed routing protocols are unable to defend against malicious attacks since all packets are sent from a source to a destination node through a single path. Therefore, cryptographic schemes have to be implemented on each sensor to ensure secure communication. However, the implementation of security mechanisms incurs cost, and therefore, a trade-off has to be maintained between the required level of security and the cost. The next chapter presents a key exchange mechanism and two lightweight encryption algorithms for resource constrained WSNs. Furthermore, a performance evaluation on real sensors is presented to provide a good understanding of the implementation platforms and the performance of the block ciphers.
Single path routing protocols are unable to address the reliability and security issues in WSNs. Thus, the presence of a malicious node on the routing path could manipulate and corrupt transmitted information between two nodes. This chapter proposes two encryption algorithms to provide end-to-end data confidentiality in WSNs. Section 5.2 proposes a Simple Lightweight Encryption Scheme (SLES) including a key exchange mechanism which uses an elliptic curve to generate a large key pool and pseudorandom bit sequences. On the other hand, Section 5.3 proposes a Chaotic-map and Genetic-operation based Encryption Algorithm (CGEA) which combines the benefits of elliptic curves, chaotic maps, and genetic operations. Both encryption algorithms are fast, robust, and computationally secure. The proposed algorithms are tested in a simulator, a emulator, and real hardware. Finally, a performance evaluation on block ciphers is presented to provide a good understanding of the trade-off between cost and security.

5.1 Introduction

WSNs aimed at various industrial, medical, and military applications necessitate research in the design of secure and energy efficient protocols. To ensure the authenticity, confidentiality, and integrity of the sensed and transmitted data, these applications require effective security mechanisms. However, security is a challenging
issue in WSNs since sensors are usually deployed in hostile environments. Moreover, limited memory and processing power, and short communication range of sensor nodes introduce several challenges when implementing cryptographic schemes in wireless environments. Therefore, WSNs require efficient encryption schemes in terms of storage space, power consumption, and operating speed.

Single path routing protocols are vulnerable to a number of security attacks such as eavesdropping, compromised node attack, and jamming attack. If a node is compromised, the adversary can reveal all cryptographic information such as secret keys and nonces. Furthermore, an adversary can eavesdrop the communication channel to capture and decrypt all packets transmitted between two nodes since the protocols use a pre-defined route to send data packets. This is a major limitation of single path routing compared to multipath routing protocols. In multipath routing, an adversary can obtain only a part of transmitted information since the packets are forwarded through multiple node-disjoint paths. In the same way, the effect of jamming attack or compromised node attack is also confined in a small area of the network in multipath routing. To minimise the damage of such security attacks, it is necessary to securely construct the routes for every SN in the network as well as to ensure the authenticity, confidentiality, and integrity of the transmitted information. To achieve the above security goals, a number of security measures should be combined such as OHC, MAC, key exchange, and encryption. This chapter proposes a key exchange and two lightweight encryption algorithms which can be used by both single path and multipath routing protocols to provide node verification and end-to-end data confidentiality in WSNs. Chapter 6 integrates the proposed security measures with a multipath routing protocol to establish a secure communication channel for both route construction and data transmission phases.

5.2 The Simple Lightweight Encryption Scheme

SLES is a symmetric encryption scheme for tiny sensor devices guaranteeing data confidentiality between source and destination nodes. The algorithm is based on a
Chapter 5. Securing Single Path Routing

pseudorandom bit sequences generated by using the points on an elliptic curve. In SLES, two nodes share a common secret key (a point on elliptic curve) to verify each other, and then start the communication process. Furthermore, this common secret key is used as the base point in generating random bit sequences to encrypt the plaintext. Thus, the proposed scheme uses the same framework for both node verification and encryption procedures. The following describes the key establishment, pseudorandom bit sequences generation, and encryption procedures of SLES.

5.2.1 Key Establishment Procedure

The key establishment phase uses an elliptic curve over a prime field of \( p \) elements to generate a large key pool for the purpose of node-verification. An elliptic curve over the prime field is an algebraic expression and is defined by the following equation:

\[
y^2 \equiv x^3 + Ax + B \quad (mod \ p)
\]  

(5.1)

where, \( A \) and \( B \) are the coefficients, and the variables \( x \) and \( y \) take the values only from the finite field within the range of prime \( p \). Given the values of these parameters, a large number of points on the curve can be generated using basic elliptic curve operations, known as point addition and point doubling [157]. In SLES, the sequence is generated by using point addition which is followed by the point doubling operation.

We assume that the elliptic curve parameters (i.e., prime field \( p \), base point \( G (x, y) \), coefficients \( A \) and \( B \)) are predistributed securely among all sensor nodes in the WSN. Now, each SN generates a list of elliptic curve points referred to as key pool by using elliptic curve operations. When a node is required to send data packets, it randomly selects a secret key \((x_i, y_i)\) from the key pool and converts it into hash code using a pre-defined hash function. Then, the hash code is shared with the destination node. The destination node retrieves the shared key by matching the received code with the hash code generated for each point of its own key pool. Upon successful retrieval of the secret key, destination node verifies the legitimacy of the
source node and sends an acknowledgement. This key is used in generation of the pseudorandom bit sequences at the next step.

5.2.2 Generation of Pseudorandom Bit Sequence

The security level of many cryptographic schemes using Linear Feedback Shift Registers (LFSRs) or chaotic maps depends on the properties of random number generation schemes such as unpredictability and unlimited period. However, the security strength of an LFSR is poor and cannot meet the demand of unpredictability [158, 159]. Similarly, some chaotic maps need high-precision floating point calculations, and thus they are not suitable for SNs. To avoid these problems, we use an elliptic curve over prime field to generate the random bit sequences.

It is assumed that the value of the following elliptic curve parameters \((A, B, \text{and } p)\) are pre-distributed among all sensors and the participating nodes share a secret key using the proposed key exchange mechanism. This key is used as the base point \((G)\) to generate the random bit sequences in SLES. Given the values of all parameters, we can generate the pseudorandom bit sequences following the procedure described in Pseudocode 5.1.

---

**Pseudocode 5.1: Pseudorandom binary sequence generation process**

**Input:** Coefficients \((A, B)\), Base Point \(G (x, y)\), Prime field \(p\)

**Output:** Binary Sequence of length \(N\) ; Initially, \(N\) equals to zero

**Steps:**

1. Generate a new point \(\bar{G} (\bar{x}, \bar{y})\) using point addition and doubling operation
2. **if** \(\bar{x} > \bar{y}\)
   
   Binary Sequence \((N) \leftarrow (\bar{x} \mod 2)\)

   **else**

   Binary Sequence \((N) \leftarrow (\bar{y} \mod 2)\)

3. \(N \leftarrow N + 1\)

4. Repeat step 1 to 3 until \(N \neq \) desired length
The randomness of the obtained binary sequence is tested using the statistical test suite developed by the National Institute of Standards and Technology (NIST) and found the sequence to be random [160]. Here is an example that shows the uniqueness of the pseudorandom bit sequence generated by using two different base points.

The following elliptic curve equation and prime field are chosen to demonstrate the example:

\[ y^2 \equiv x^3 - x + 188 \mod 751 \]  \hspace{1cm} (5.2)

where, \( A = -1 \), \( B = 188 \), and \( p = 751 \). The important property of an elliptic curve is that the base point, \( G (0, 376) \) will generate the ‘pseudorandom bit sequence’ which would be different than the ‘pseudorandom bit sequence’ generated by the base point, \( G (1, 376) \). The following shows the first 32 bits of the pseudorandom bit sequence obtained for two different base points.

\[
\begin{align*}
G (0, 376) & - 01111000 11110111 10100100 10011111 \\
G (1, 376) & - 00110000 01000001 11011000 01001111
\end{align*}
\]

5.2.3 The Encryption Procedure

The proposed encryption scheme is very simple and involves only the bitwise XOR operation as shown in Figure 5.1. The random bit sequences obtained at the previous stage works as one time passwords in SLES. At first step, the plaintext is converted into binary sequence by mapping the characters into their corresponding ASCII codes. Then, the binary sequence is xor-ed with the pseudorandom bit sequence to generate the ciphertext. We perform the XOR operation because the additive cipher is more secure when the key-stream is random and as long as the plaintext [161]. The decryption process is same as the encryption procedure, where the ciphertext is xor-ed with the random bit sequences to obtain the plaintext.
5.2.4 Security Analysis

To evaluate the security strength of the proposed encryption algorithm, a number of security tests are performed. The aim of these tests is to measure the following properties: i) independence of the plaintext and the ciphertext, ii) key strength analysis, and iii) cryptographic analysis. The following is a brief discussion on the results obtained from the tests.

- **Independence of Plaintext and Ciphertext**: This test aims to find randomness of large ciphertext blocks produced from the patterned plaintext and fixed keys. The following tests are implemented on the ciphertext generated by SLES, and it is expected that the output value of each test must be greater than 0.01.

  1. **Frequency Test**- The purpose of this test is to determine whether the number of ones and zeros in \( n \)-bit sequence of the ciphertext is as expected based on the chi-square distribution. The results obtained in this test is greater than 0.41 in each test case for 1000 trials.

  2. **Runs Test**- This test is performed to determine whether the number of ones and zeros of various lengths is as expected for a random sequence. It mainly shows whether the oscillation between ones and zeros is too slow or
too fast. A number of runs tests is performed on different bit sequences of the ciphertext and found the probability value (P-value) within the range of 0.72 to 0.97. This results also indicate the fast oscillation in the bit string.

3. Linear Complexity Test- The objective of this test is to find whether the sequence is complex enough to be considered as random. If the length of a linear feedback shift register is too short, then the sequence is not random. Our test results ensure the randomness of the bit sequences as the P-value obtained in tests is greater than 0.67 for each trial.

- Related-key Analysis: Related-key cryptanalysis is a type of security attack based on decrypting the ciphertext with various similar (but not identical) keys and analysing the difference in outputs. This attack is common in advanced cryptosystem, like AES and WEP [162]. In SLES, the parameters of elliptic curve equation ($A$ and $B$), prime number ($p$), and base point $G$ ($x, y$) are the primary keys. Related-key cryptanalysis results show that it is not possible to generate an identical pseudorandom bit sequences if any of the above values is not same. Thus, it is impossible to decrypt the ciphertext without knowing exact value of each parameter used in the encryption process. Let a message be encrypted with the following values: $A = 1$, $B = 188$, $p = 751$ and $G = (0, 376)$. Then, to retrieve the original plaintext, the ciphertext is decrypted using $G (1, 376)$ while other values are unchanged. The result shows that it could not recover any partial segment of the plaintext. Similar results are obtained when the value of coefficients or prime number is slightly changed.

- Cryptanalysis: Finally, the proposed encryption scheme is tested to evaluate the security strength of the cryptosystem against some common attacks such as brute force attack, statistical analysis attack, and timing attack.

1. Brute Force Attack- The security of elliptic curve cryptosystems relies on the discrete logarithm problem, and the best known algorithm to solve those problems requires an exponential time. Pohlig-Hellman and
Pollard-Rho algorithm are two well known attacks which can solve the discrete logarithm problem in exponential time. However, these attacks involve huge computational costs in terms of the CPU cycles and memory. NIST recommends to use a 256-bits key for secure data transmission although the ECC scheme broken to date had a 112-bits key for the prime field.

2. **Statistical Analysis**- Statistical analysis or frequency analysis is the study of the frequency of letters or common characteristics of words in the ciphertext. Generally, the message of plaintext is text or other information, and there is a characteristic distribution and commonly used combination of letters, such as, in English language, i) the letter e, t, a, o have a high frequency of use, while letters q, z, x do not; ii) TH, ER, ON and AN are the most common pairs of letters, and iii) in English word, q is always followed by u. An English article of more than 10,000 words is used as the plaintext to generate the corresponding ciphertext. However, the ciphertext does not have any statistical features: all characters are randomly distributed and they do not follow any particular order. Moreover, a character is represented by different cipher-code most of the time in the ciphertext. Therefore, it is too hard to find a correlation in the ciphertext.

3. **Timing Attack**- Timing attack is not possible in SLES due to data independent behavioural characteristics of the algorithm. Moreover, the binary sequence used in encryption varies each time, hence, it is quite impossible to derive any statistical correlation of timing information.

### 5.2.5 Performance Analysis

The proposed encryption scheme including key setup phase is implemented in wireless sensor mote called MICA2 composed of a microprocessor (ATmega128L) operating at 7.3728 MHz, 128 KB program memory, and 4 KB data memory. The mote supports an event driven operating system commonly known as TinyOS and
a high level programming language based on components called nesC. Furthermore, the algorithm is evaluated in an emulator called ATEMU, used to perform high fidelity large scale sensor network emulation studies in a controlled environment. NIST recommended 128-bit elliptic curve domain parameters over prime field have been used in the experiments. RC5 and non-optimised Skipjack algorithms are also implemented in the TinyOS environment and the results are compared with SLES.

- **Memory Efficiency**: The nesC source code compilation process generates a report indicating memory occupation (for ROM and RAM) in bytes for MICA2 mote. Table 5.1 represents the amount of memory occupied by RC5, Skipjack, and SLES.

<table>
<thead>
<tr>
<th>Memory</th>
<th>Skipjack</th>
<th>RC5</th>
<th>SLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM</td>
<td>7218</td>
<td>6396</td>
<td>5326</td>
</tr>
<tr>
<td>RAM</td>
<td>292</td>
<td>376</td>
<td>542</td>
</tr>
<tr>
<td>.bss</td>
<td>14</td>
<td>14</td>
<td>54</td>
</tr>
<tr>
<td>.text</td>
<td>278</td>
<td>362</td>
<td>488</td>
</tr>
<tr>
<td>.data</td>
<td>7204</td>
<td>6382</td>
<td>5272</td>
</tr>
</tbody>
</table>

It can be seen that SLES is more memory-efficient compared to RC5 and Skipjack. The flash memory (ROM) required by SLES is lower than that of RC5 and Skipjack but it occupies more RAM than the other two schemes. However, the total memory (ROM and RAM together) consumed by the proposed algorithm is 5868 bytes, whereas the amount is 6772 bytes for RC5 and 7510 bytes for Skipjack.

- **Operation Time**: Operation speed indicates the time efficiency which is an important factor for performance evaluation. ATEMU is used to calculate the total CPU cycles required to encrypt 32 bytes data for MICA2 sensor mote, whereas TOSSIM is used to measure the total encryption time in milliseconds for RC5, Skipjack, and SLES. Table 5.2 shows the outcomes obtained in the experiments.
Table 5.2: CPU consumption and elapsed time to encrypt 32 bytes data

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>CPU Cycles</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skipjack</td>
<td>91224</td>
<td>12.353</td>
</tr>
<tr>
<td>RC5</td>
<td>48709</td>
<td>6.595</td>
</tr>
<tr>
<td>SLES</td>
<td>45839</td>
<td>6.207</td>
</tr>
</tbody>
</table>

The results indicate that SLES performs better in terms of the CPU elapsed time (6.207 ms) and the CPU cycles (45839). The number of CPU cycles required by RC5 is a bit higher compared to SLES, whereas the number is about two times greater than that of SLES. The elapsed time to encrypt 32 bytes data also shows the same results.

- **Energy Efficiency:** Energy consumption represents the cost of computational complexity of an algorithm, and it is proved that faster block ciphers consume less energy. In this experiment, we used the dedicated power consumption estimation module, called PowerTOSSIM to measure the total amount of energy required to encrypt and successfully send data packets by the MICA2 sensor mote. The encryption energy represents the amount of energy consumed to encrypt 32 bytes data, whereas the communication energy is the total energy consumed to encrypt 32 bytes data as well as successfully transmit data to the destination node. Figure 5.2 indicates that SLES consumes less energy than RC5 and Skipjack. The total energy dissipation by the RC5 cipher for both encryption and communication is slightly higher than that of SLES, whereas it is almost double for the Skipjack cipher. SLES reduces a significant amount of energy when large volumes of data is encrypted.

### 5.2.6 Discussion

The main objective of the elliptic curve based pseudorandom bit sequence generator is to avoid the floating point arithmetic operations involved in the random sequence generation process. The algorithm yields different pseudorandom bit sequences for each session to ensure independent behavioural characteristics of the cipher. Due to
the ability of generating longer bit sequences, SLES is suitable for large volume data encryption such as image, audio, and video. Furthermore, it can be implemented in the form of both block cipher and stream cipher. However, the proposed scheme has a drawback. A 128-bit elliptic curve is used to implement SLES in the experiments. If a 256-bit elliptic curve is used for an enhanced level of security, it will result in additional computational cost and memory usage. To overcome this problem, another lightweight encryption algorithm, Chaotic-map and Genetic-operations based Encryption Algorithm has been proposed.

5.3 The Chaotic-map and Genetic-operation based Encryption Algorithm

CGEA is a lightweight block cipher based on a chaotic map and genetic operations. The proposed encryption scheme employs elliptic curve points to verify the communicating nodes and also as one of the chaotic map parameters to generate the pseudorandom bit sequences. This sequence is used in XOR, mutation and crossover operations in order to encrypt the data blocks. The algorithm includes a number
of unique benefits: i) it uses the discrete chaotic map, which supports a wider data range with low computational cost. Most of the encryption schemes use fixed chaotic map parameters to generate the random bit sequences, but CGEA uses random values of $x$ and $y$ for every new session; ii) the proposed crypto-system makes different pseudorandom bit sequences for different sessions, and thus preserves independent behavioural characteristics of the algorithm; iii) the scheme is more efficient compared to Skipjack, AES, LED, TWINE, and BCC in terms of CPU consumption and encryption time; iv) the proposed encryption algorithm is suitable for both text and image encryption. From the application point of view, it is desirable for the cryptosystem to protect confidential information not only in text form but also in image form. Image data differ from text due to intrinsic features, such as strong correlation between adjacent pixels and high redundancy. Hence, the encryption scheme should be robust, fast, and computationally secure. Experimental results show that the proposed block cipher ensures all these properties.

The proposed encryption scheme is divided into three phases: a) key establishment, b) pseudorandom bit sequences generation, and c) encryption. Each phase is described in detail below.

### 5.3.1 Key Establishment Phase

CGEA uses the same procedure to exchange the secret key i.e., an elliptic curve point $(x_i, y_i)$ between two nodes as described in section 5.2.1. However, in CGEA, this secret key is used in the N-logistic tent map with other parameters to generate the random bit sequences. For every new session, the communicating nodes selects a new key in order to generate different pseudorandom bit sequences.

### 5.3.2 Generation of Pseudorandom Bit Sequence

This phase involves the generation of pseudorandom bit sequences using a chaotic function. Most of the chaotic maps involve high-precision floating point calculations to produce a sequence of random floating-point numbers which are not suitable for
resource limited SNs. However, the advantage of using the N-logistic tent map is that it can deal with integer parameters, and thus simplifies the computation process in SNs. We have investigated the randomness of the derived binary sequences using the test code developed by the NIST and found the sequence to be random. The following equations define the chaotic functions used to generate the pseudorandom bit sequences in the proposed encryption scheme.

\[
\begin{align*}
    x_{n+1} &= \mu x_n (N - x_n/m)/N - y_n/2 \\
    y_{n+1} &= \beta (N - |N - y_n|)
\end{align*}
\]  

(5.3)

where \( x \in (0, m\times N) \), \( \mu \in [0, 4] \), \( y \in (0, 2\times N) \), \( \beta \in [1, 2] \), \( N = 2^K \), and \( m = 2^k \) with integers \( K \) and \( k \) [163]. The seed key is the set \{ \( x_i, y_i, m, N, \mu, \beta \) \}, where, the values of \( m, N, \mu, \) and \( \beta \) are predistributed in sensor nodes, and the initial values of \( x_i \) and \( y_i \) are exchanged through the key establishment phase as explained in section 5.2.1.

### 5.3.3 The Encryption Process

The overall encryption process is shown as a block diagram presented in Figure 5.3, where the symbols ‘M’ and ‘XO’ denote mutation and crossover operations respectively. Confusion and diffusion are two general principles that guide the design of a block cipher. Confusion is achieved by obscuring the relationship between the ciphertext and the symmetric key as best as possible. On the other hand, diffusion is achieved by dissipating the redundancy of the plaintext through spreading it over the ciphertext. The proposed cryptographic scheme implements three different operations: XOR, mutation, and crossover. The additive cipher XOR is secure when the key-stream is as long as the plaintext. On the other hand, mutation is a process of flipping one or multiple bits in a given bit string. Crossover is a process of taking two parent bit strings and producing corresponding child bit strings by interchanging selected parts of the bit strings between the parents. These two genetic operations are used in the genetic algorithm to generate a new population from the existing one [164]. In CGEA, the mutation and crossover genetic operations are used as tools
for introducing diffusion and confusion properties in the ciphertext. The mutation process is applied to create random diversity (diffusion) in the ciphertext, whereas the crossover operation is used to change the order of the mutated text or image data (confusion). The main benefit of using genetic operations is that they introduce relatively fair diversity in the ciphertext. Below, we describe the encryption procedure with examples.

We first divide the pseudorandom bit sequences generated by the chaotic map into 256-bit blocks, and each block is divided into two sub-blocks of 128 bits. Then, we calculate the number of 1’s in each byte as well as the sum of 1’s for each two consecutive bytes in the other half of the pseudorandom bit sequences. After that, we convert the plaintext into their corresponding binary codes and group them into blocks of 128-bits. This block is XORed with the first sub-block of the pseudorandom bit sequences. Then, the mutation is performed on each byte of the XORed binary codes using the total number of 1’s in each byte as the starting index of the mutation.
process. For example, if the number of 1’s in the first byte of the second sub-block of a pseudorandom bit sequences is 7, we mutate the 7\textsuperscript{th} and 8\textsuperscript{th} number bits in the first byte of the XORed plaintext. Thereafter, the crossover operation is executed on the mutated plaintext as shown in Figure 5.3 to generate the ciphertext. At this step, we take four consecutive bytes from the mutated plaintext and then perform crossover operations according to the number of 1’s in the second sub-block of the pseudorandom bit sequence. For example, let $B_1-B_2-B_3-B_4$ be four successive bytes of the mutated binary codes. We compare the sum of 1’s ($\text{sum1}$) in the first two bytes in the pseudorandom bit sequences with that of the next two consecutive bytes ($\text{sum2}$). If $\text{sum1}$ is greater than $\text{sum2}$, then the crossover is performed from left to right (i.e., 1 to $\text{sum1}$), otherwise it is done from right to left (i.e., 32 to $\text{sum2}$). This crossover is done repeatedly (e.g., $B_1-B_2-B_3-B_4$, $B_2-B_3-B_4-B_5$, \ldots, $B_{14}-B_{15}-B_{16}-B_1$) so that each byte in the mutated plaintext performs the crossover operation at least twice. The decryption procedure is simply the inverse of the encryption process.

5.3.4 Concrete Security Evaluation

An encryption scheme is provably secure if the advantage of any computationally bounded adversary is negligible. We have used concrete security analysis to obtain an accurate estimate of computational complexities of adversarial activities for our proposed encryption algorithm. In this section, we present the formal description of the proposed cryptographic scheme, mathematical notation of indistinguishability under a chosen-plaintext attack (IND-CPA), and IND-CPA security analysis.

5.3.4.1 Formal Description

A symmetric encryption algorithm is defined by a family of functions such as $F$: $\text{Keys}(F) \times \text{Dom}(F) \rightarrow \text{Range}(F)$. For $K \in \text{Keys}(F)$, $F_K : \text{Dom}(F) \rightarrow \text{Range}(F)$ can be defined as $\forall x \in \text{Dom}(F) : F_K(x) = F(K, x)$. Thus, our proposed encryption scheme with $\text{Keys}(F) = \{0, 1\}^{256}$ and $\text{Dom}(F) = \text{Range}(F) = \{0, 1\}^{128}$ can be expressed as, $F : \{0, 1\}^{256} \times \{0, 1\}^{128} \rightarrow \{0, 1\}^{128}$ where, the mode of operation
over \( F \) with a random starting point is a stateless block cipher. The encryption and decryption algorithms are presented in Pseudocode 5.2. The starting point \( S[0] \) is used to define a set of values on which \( F_K \) is applied to generate a “pseudorandom bit sequence” of desired length. Then, the sequence is subdivided into two parts (\( P[i]^L \) and \( P[i]^R \)) to perform XOR(\( \oplus \)), mutation(\( \bar{\mu} \)) and crossover(\( \otimes \)) operation as shown in Pseudocode 5.2.

Pseudocode 5.2: Encryption and Decryption Procedure

1: Procedure \( \text{ENC}_K(M) \) | 1: Procedure \( \text{DEC}_K(M) \)
2: \( M[1] \ldots M[n] \leftarrow M \) | 2: \( C[1] \ldots C[n] \leftarrow C \)
3: \( S[0] \leftarrow \{0, 1\}^l \) | 3: \( S[0] \leftarrow \{0, 1\}^l \)
4: for \( i = 1 \ldots n \) do | 4: for \( i = 1 \ldots n \) do
5: \( P[i] \leftarrow F_K(S[0], i) \) | 5: \( P[i] \leftarrow F_K(S[0], i) \)
6: \( CT_1[i] \leftarrow P[i]^L \oplus M[i] \) | 6: \( M_1[i] \leftarrow C[i] \otimes P[i]^R \)
7: \( CT_2[i] \leftarrow CT_1[i] \bar{\mu} P[i]^R \) | 7: \( M_2[i] \leftarrow M_1[i] \bar{\mu} P[i]^R \)
8: \( C[i] \leftarrow CT_2[i] \otimes P[i]^R \) | 8: \( M[i] \leftarrow P[i]^L \oplus M_2[i] \)
9: end for | 9: end for
10: return \( C \) | 10: return \( M \)
11. end procedure | 11. end procedure

5.3.4.2 Indistinguishability under Chosen-Plaintext Attack

Suppose an adversary is given a sequence of ciphertexts \( (C_1 \ldots C_n) \) where \( C_i \) is either an encryption of \( M_{0,i} \) or \( M_{1,i} \) for all \( 1 \leq i \leq n \). Now, the goal of the adversary is to generate the sequence of ciphertexts and guess whether \( M_{0,i} \ldots M_{0,n} \) were encrypted or \( M_{1,i} \ldots M_{1,n} \) were encrypted. The encryption scheme is considered secure if the adversary finds it hard to distinguish which of the two message sequences correspond to the cipher, \( C_i \).

Let us now formalise an attack scenario. Let \( \pi = (K, \xi, D) \) be a symmetric encryption scheme. An adversary ‘\( A \)’ is a program that has access to an oracle known as LR (left or right) oracle. ‘\( A \)’ can input any pair of equal-length messages and the oracle will return a ciphertext. For \( K \in \text{Key} \), let \( \text{Real}_K \) be the left oracle that on input \( M \) returns \( C \leftarrow \xi_K(M) \) and \( \text{Fake}_K \) be the right oracle that on input
Chapter 5. Securing Single Path Routing

M returns $C \xrightarrow{\xi} \xi_K(0^{|M|})$. Now, the ind-cpa advantage of $A$, i.e., the success of the adversary in breaking the encryption scheme can be defined by the difference in probabilities of the two worlds as follows [165]:

$$Adv_{ind-cpa}^\pi(A) = Pr[Real^A_\pi \Rightarrow 1] - Pr[\text{Fake}^A_\pi \Rightarrow 1]$$ (5.4)

If $Adv_{ind-cpa}^\pi(A)$ is very small (i.e., close to zero), it indicates that ‘$A$’ is doing poorly and $F$ resists the attacks that $A$ is mounting. On the other hand, if $Adv_{ind-cpa}^\pi(A)$ is large (i.e., close to one), it means that ‘$A$’ is doing well and $F$ is not secure. The next sub-section uses the above analysis to test the security of the proposed scheme.

5.3.4.3 IND-CPA Security of the Encryption Scheme

Let $F : \{0, 1\}^{256} \times \{0, 1\}^{128} \rightarrow \{0, 1\}^{128}$ be our proposed block cipher. Let $A$ be an adversary that executes in time $t$ and performs at most $q$ queries, which means totaling at most $\sigma$ $n$-bit blocks. Then, according to [166], there exists an adversary $B$, attacking the pseudorandom function (PRF) security of $F$ such that

$$Adv_{ind-cpa}^\pi(A) \leq Adv_{prf}^F(B) + \frac{0.5\sigma^2}{2^n}$$ (5.5)

Although CGEA cipher is provably secure, there is a possibility that a collision may produce ‘overlaps’ in the pseudorandom bit sequences. We assume that the best attack against the PRF security of our proposed block cipher is a birthday attack since the attack is used to find a collision between random attack attempts. As mentioned before, $F$ is our proposed block cipher that takes 128 bits plaintext to generate corresponding ciphertext. Suppose we want to encrypt $q = 2^{20}$ messages, where each message is $2^{15}$ bits long, thus in total, $2^{35}$ bits have to be encrypted resulting in a total of $2^{28}$ number of blocks ($\sigma$). Let $A$ be an adversary that cannot do better than mounting a birthday attack ($B$) against the block cipher $F$. It means that the advantage of the attack $Adv_{prf}^F(B) \not\geq \sigma^2/2^{128}$. Under this condition, the IND-CPA security of our proposed encryption scheme is:

$$Adv_{ind-cpa}^\pi(A) = \sigma^2/2^{128} + 0.5\sigma^2/2^{128} = 1.5(2^{56}/2^{128}) \leq 1/2^{71},$$

which is a very small
number indeed. It indicates that the cipher is secure under the assumption that the best attack on the PRF security of the proposed block cipher is a birthday attack.

5.3.5 Security Analysis and Test Results

We have tested our proposed encryption scheme against various security attacks. Here, we describe some of the important security analysis results including key-space analysis, statistical analysis, differential attack analysis, information entropy analysis, and known-plaintext and ciphertext-only attack analysis. The experiments are performed on two gray-level images with a size of $128 \times 128$ using the MATLAB simulator.

5.3.5.1 Key-space Analysis

Key space size is determined by the total number of different keys used in the encryption scheme. The key-space of a good encryption algorithm should be large enough to make brute-force attacks infeasible. In our proposed encryption scheme, the set of secret parameters is $\{ x_i, y_i, \mu, \beta, m, N \}$, where, $x_i$, $y_i$, $\mu$, $\beta$, $m$, and $N$ are integer values, having the following range: $\{ x_i, y_i, \mu, \beta, N \} \in [1, 2^{128}]$ and $m \in [1, 2^{64}]$. Therefore, the complete key-space of the proposed encryption scheme is $5.087 \times 10^{135} \approx 2^{448}$. Hence, we conclude that brute force attack is not feasible for such a large key space.

5.3.5.2 Statistical Analysis

We have performed several statistical analysis to test the robustness of CGEA. From the experiments, we have found that the encrypted information in the cipher image is nearly uniformly distributed. Here, we present the results of the histogram analysis and correlation analysis of two adjacent pixels in cipher images.
5.3.5.3 Histogram Analysis

An image histogram plots the pixels at each gray scale level in order to illustrate the distribution of pixels in that image. It is expected that the distribution of pixels in the cipher image should hide the redundancy of the original image and should not leak any information about the plain image or the relationship between the plain image and the cipher image. We have calculated and analysed the histograms of two encrypted images as well as original images consisting of different contents. The histograms of the plain images (Figure 5.4(a) and 5.5(a)) and their corresponding cipher images (Figure 5.4(c) and 5.5(c)) produced by the proposed scheme are depicted in Figure 5.4(b), 5.5(b); and 5.4(d), 5.5(d) respectively. Figures 5.4(d) and

![Figure 5.4: Histograms of the plain and cipher images: (a) Plain image (Lena.png) (b) Histogram of Lena.png (c) Cipher image (Lena_enc.png) (d) Histogram of Lena_enc.png](image-url)
5.3.5.4 Correlation of Two Adjacent Pixels

The pixels in an ordinary image are highly correlated with their adjacent pixels either in horizontal, vertical, diagonal or anti-diagonal directions. However, a secure encryption scheme should maintain sufficiently low correlation in the adjacent pixels in the cipher image. To compare the correlations of the adjacent pixels, we have calculated the correlation between two vertically, two horizontally, two diagonally,
and two anti-diagonally adjacent pixels in the plain and cipher images respectively. We have performed the following procedure to find out the correlation between two adjacent pixels. We have randomly selected 1000 pairs of adjacent pixels from the plain images. Then, we have calculated their correlation coefficient using the following formulas:

\[
cov(x, y) = E((x - E(x))(y - E(y)))
\]  
(5.6)

\[
r_{xy} = \frac{cov(x, y)}{\sqrt{\text{var}(x)\text{var}(y)}}
\]  
(5.7)

\[\text{Figure 5.6: Correlation of two horizontally adjacent pixels: (a) Plain image (Lena.png) (b) Cipher image (Lena_enc.png) (c) Plain image (Mona.gif) (d) Cipher image (Mona_enc.gif)}\]
where $x$ and $y$ are gray-levels of two adjacent pixels in the image. Figure 5.6(a) and Figure 5.6(c) show the correlations of two horizontally adjacent pixels in the original images in Figure 5.4(a) and Figure 5.5(a) respectively, whereas Figure 5.6(b) and Figure 5.6(d) represent the correlation of two horizontally adjacent pixels in the corresponding cipher images in Figure 5.4(c) and Figure 5.5(c) respectively. From the figures, it can be seen that the two horizontally adjacent pixels in the plain images are highly correlated but the correlation between the two adjacent pixels in the cipher images is negligible. Similar results are obtained for the vertical, diagonal, and anti-diagonal adjacent pixels as shown in Table 5.3. The adjacent pixels in plain images maintain a high level of correlation ranging from 0.7093 to 0.9473, whereas the correlation coefficients of the adjacent pixels in the cipher images vary from 0.0019 to 0.0356. The results indicate that it is not possible to retrieve the statistical information of the plain images from the cipher images.

<table>
<thead>
<tr>
<th>Input Images</th>
<th>Horizontal</th>
<th>Vertical</th>
<th>Diagonal</th>
<th>Anti-diagonal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena.png</td>
<td>0.8960</td>
<td>0.9473</td>
<td>0.8647</td>
<td>0.8907</td>
</tr>
<tr>
<td>Lena_enc.png</td>
<td>0.0027</td>
<td>0.0019</td>
<td>0.0070</td>
<td>0.0034</td>
</tr>
<tr>
<td>Mona.gif</td>
<td>0.7798</td>
<td>0.7719</td>
<td>0.7093</td>
<td>0.7343</td>
</tr>
<tr>
<td>Mona_enc.gif</td>
<td>0.0197</td>
<td>0.0356</td>
<td>0.0061</td>
<td>0.0280</td>
</tr>
</tbody>
</table>

### 5.3.5.5 Differential Attack Analysis

To test the impact of a one-pixel change in the original image, two common measures are used: the number of pixels change rate (NPCR) and the unified average changing intensity (UACI). Let $C_1$ and $C_2$ be two cipher images, whose corresponding plain images have only one pixel difference. We define the gray values of the pixels at grid $(i, j)$ in $C_1$ and $C_2$, as $C_1(i, j)$ and $C_2(i, j)$, respectively. We declare a bipolar array $D$ with the same size as image $C_1$ or $C_2$. The value of $D(i, j)$ is determined
Chapter 5. Securing Single Path Routing

by $C_1(i, j)$ and $C_2(i, j)$ as follows; if $C_1(i, j) = C_2(i, j)$ then $D(i, j) = 1$, otherwise $D(i, j) = 0$. The NPCR is defined by the following formula:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \quad (5.8)$$

where $W$ and $H$ are the width and height of $C_1$ or $C_2$. Similarly, the UACI is defined by the following equation:

$$UACI = \frac{1}{W \times H} \left[ \sum_{i,j} \left| C_1(i,j) - C_2(i,j) \right| \right] \times 100\% \quad (5.9)$$

The NPCR calculates the percentage of different pixel numbers between two images and the UACI calculates the average intensity of differences between two images. To calculate the NPCR and UACI value for our proposed algorithm, we have changed one pixel value of the two plain images (Figure 5.4(a) and Figure 5.5(a)) and generated the corresponding cipher images. The results of the NPCR and UACI tests are shown in Table 5.4. From the results, it can be seen that CGEA passes both the NPCR and UACI randomness tests since the obtained values are very close to ideal values of NPCR (99.6%) and UACI (33.4%). The results indicate that a minor change in the plain images results in a significant change in the cipher images. Therefore, the proposed encryption schemes has good ability to defend against the differential attack.

<table>
<thead>
<tr>
<th>Input Images</th>
<th>NPCR</th>
<th>UACI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena_enc.png</td>
<td>99.676</td>
<td>33.462</td>
</tr>
<tr>
<td>Mona_enc.gif</td>
<td>99.621</td>
<td>33.422</td>
</tr>
</tbody>
</table>

5.3.5.6 Information Entropy Analysis

Information entropy presents the degrees of uncertainty in the system and can be used to analyse the indeterminateness of an encryption scheme. We can formally
define the entropy, \( H(m) \) of any message \( m \) as follows:

\[
H(m) = \sum_{i=1}^{N} p(m_i) \cdot \log_2 \frac{1}{p(m_i)}
\]  \hspace{1cm} (5.10)

where \( p(m_i) \) denotes the probability of the symbol \( m_i \) and \( N \) is the total number of symbols. If the output of a cipher emits \( 2^n \) symbols, then the entropy should be \( n \). As an example, the ideal entropy of a 256-gray scale image must be 8 since the pixel elements have \( 2^8 \) possible values. If the entropy value is less than 8, there exists a certain degree of predictability, which threatens the security of the encryption algorithm.

Table 5.5 shows the entropies for both plain images and cipher images. It can be seen that the entropy values of the cipher images are much closer to the expected value of 8. This means that the chance of information leakage is negligible and CGEA is secure against the entropy attack.

**Table 5.5: Information entropy of plain and cipher images**

<table>
<thead>
<tr>
<th>Input Images</th>
<th>Plain image</th>
<th>Cipher image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>7.4102</td>
<td>7.9988</td>
</tr>
<tr>
<td>Mona</td>
<td>7.3499</td>
<td>7.9884</td>
</tr>
</tbody>
</table>

### 5.3.5.7 Known Plaintext and Ciphertext only Attack

The known plaintext attack (KPA) is an attack based on having samples of both plaintext and corresponding ciphertext. Now, using this information, the attacker tries to find the secret key used in the encryption and decryption procedures. However, KPA is not feasible in CGEA since different plain images are encrypted using a different key stream. Hence, it is not possible to obtain useful information by encrypting any special image because the resultant cipher depends upon a random number of operations on the basis of the key stream.
Chapter 5. Securing Single Path Routing

The ciphertext only attack (COA) is an attack model used in cryptanalysis when the attacker has access only to a set of ciphertext. For a given set of ciphertext, if the attacker can determine corresponding plaintext then COA is successful. Suppose an adversary performs exhaustive search on the first 1024 bits of a cipher image to retrieve one-sixteenth segment of the plain image. The possible combinations will then be a number of $2^{1024} \approx 1.797 \times 10^{308}$, which indicates that the COA attack on this proposed encryption scheme is infeasible.

5.3.6 Performance Comparison

We implemented the proposed encryption scheme in a MICA2 sensor mote as described in Section 5.2.5. The computational cost of key establishment phase is not considered in the experiments since this process is done only once during the setup phase. The AES, non-optimised Skipjack, LED, TWINE, SLES-256, and BCC algorithms are also implemented in TinyOS environment and the results are compared with the proposed cryptographic scheme as shown in Table 5.6.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CPU Cycles</th>
<th>Time (ms)</th>
<th>RAM (bytes)</th>
<th>ROM (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skipjack</td>
<td>91224</td>
<td>12.353</td>
<td>292</td>
<td>7218</td>
</tr>
<tr>
<td>AES</td>
<td>68512</td>
<td>9.287</td>
<td>324</td>
<td>6994</td>
</tr>
<tr>
<td>LED</td>
<td>589652</td>
<td>78.972</td>
<td>378</td>
<td>5970</td>
</tr>
<tr>
<td>TWINE</td>
<td>128896</td>
<td>17.477</td>
<td>384</td>
<td>5280</td>
</tr>
<tr>
<td>SLES-256</td>
<td>82397</td>
<td>11.154</td>
<td>842</td>
<td>7856</td>
</tr>
<tr>
<td>BCC</td>
<td>91286</td>
<td>12.547</td>
<td>976</td>
<td>6240</td>
</tr>
<tr>
<td>CGEA</td>
<td>62396</td>
<td>8.547</td>
<td>542</td>
<td>5326</td>
</tr>
</tbody>
</table>

We have used ATEMU to get the total CPU cycles required to encrypt 32 bytes data by the MICA2 sensor mote. On the other hand, using TOSSIM, we have calculated the memory consumption and total encryption time in milliseconds for the Skipjack, AES, LED, TWINE, BCC, SLES-256, and CGEA cipher. The results in
the table indicate that our proposed algorithm performs better in terms of the CPU elapsed time (8.547 ms) using only 62396 CPU cycles. For AES, Skipjack, SLES-256, and BCC, the number of CPU cycles and encryption time is higher compared to our scheme, while the elapsed time and required CPU cycles are almost double for TWINE and nine times higher for the LED cipher. In case of memory consumption, it can be seen that TWINE is more efficient than the other algorithms. Although CGEA uses marginally more memory compared to TWINE, it is less than that of Skipjack, AES, LED, SLES-256, and BCC.

5.3.7 Discussion

CGEA is a fast, provably secure, and robust block cipher proposed for sensor networks. The encryption scheme incorporates the benefits of elliptic curve operations, chaotic map, and genetic operations to provide node verification and data confidentiality. The cipher randomly selects different secret keys \((x_i, y_i)\) rather than the fixed parameters for every new session. This mechanism makes the cipher hard to break for adversaries. Another advantage of the proposed scheme is that it can be used to encrypt both text and image data. Thus, CGEA is more suitable for use in multimode sensors. Theoretical analysis and experimental results show that the proposed block cipher is provably secure and is more resource efficient in terms of resource consumption.

5.4 Performance Evaluation on Hardware

The implementation of security policies has to maintain a trade-off between cost and performance. For example, many WSN applications require complex cryptographic algorithms to provide an enhanced level of security. However, the cost increases as powerful SNs are required to implement the crypto-system. Therefore, it is necessary to clearly understand the relationship between the implementation cost and effectiveness. Table 5.7 presents a comparative view of costs and hardware specifications of a number of sensor motes. The table shows that EZ430-RF2500 and Arduino Pro
motes are less costly but they also have less memory. Hence, performance evaluation of cryptographic schemes on low-cost SNs is necessary to examine the feasibility of cost-effective platforms.

Table 5.7: Costs and hardware specifications (2014)

<table>
<thead>
<tr>
<th>Sensor Motes</th>
<th>Price (US$) (incl.)</th>
<th>Package</th>
<th>Microcontroller</th>
<th>Bus (bit)</th>
<th>Clock (MHz)</th>
<th>RAM (KB)</th>
<th>Flash (KB)</th>
<th>EEPROM (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHIMMER 226</td>
<td>2 boards 26</td>
<td>MSP430F16116</td>
<td>4-8</td>
<td>10</td>
<td>48</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wasp mote</td>
<td>168</td>
<td>ATmega12818</td>
<td>14</td>
<td>8</td>
<td>128</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IRIS</td>
<td>134</td>
<td>ATmega12818</td>
<td>8</td>
<td>8</td>
<td>640</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TelosB</td>
<td>102</td>
<td>MSP430F16116</td>
<td>4-8</td>
<td>10</td>
<td>48</td>
<td>1024</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mica2</td>
<td>99</td>
<td>ATmega128L8</td>
<td>8</td>
<td>4</td>
<td>128</td>
<td>512</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EZ430-RF2500</td>
<td>40</td>
<td>MSP430F22746</td>
<td>16</td>
<td>1</td>
<td>32</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arduino Pro (328)</td>
<td>25</td>
<td>nRF24L01 radio</td>
<td>ATmega328</td>
<td>8</td>
<td>16-2</td>
<td>32</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

This section presents an experimental evaluation of cryptographic algorithms based on sensor hardware. A number of block ciphers namely Skipjack, XXTEA, RC5, AES, and CGEA are implemented in both Mica2 and Arduino Pro mote platforms in order to compare the memory efficiency, computational cost, and operation time. Based on the results, some critical insights are also provided which might be useful to select the best cryptographic algorithm and implementation platform. Furthermore, this performance evaluation will help to assess the security strength and feasibility of the proposed CGEA cipher.

5.4.1 Implementation Environment

The Skipjack, XXTEA, RC5, AES, and CGEA ciphers are briefly described in this section. Table 5.8 lists the parameters used for the block ciphers in our experiments. CGEA performs iterative genetic operations instead of rounds, whereas Skipjack uses
32 rounds with an 80-bits key. XXTEA, RC5, and AES-256 executes for 14 rounds, whereas AES-128 runs for 10 rounds only.

Table 5.8: Cipher parameters used in experiments

<table>
<thead>
<tr>
<th>Block Ciphers</th>
<th>Key Length (bits)</th>
<th>Rounds</th>
<th>Block Length (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skipjack</td>
<td>80</td>
<td>32</td>
<td>64</td>
</tr>
<tr>
<td>XXTEA</td>
<td>128</td>
<td>14</td>
<td>64</td>
</tr>
<tr>
<td>RC5</td>
<td>128</td>
<td>14</td>
<td>64</td>
</tr>
<tr>
<td>AES-128</td>
<td>128</td>
<td>10</td>
<td>128</td>
</tr>
<tr>
<td>AES-256</td>
<td>256</td>
<td>14</td>
<td>128</td>
</tr>
<tr>
<td>CGEA</td>
<td>256</td>
<td>N/A</td>
<td>128</td>
</tr>
</tbody>
</table>

5.4.1.1 Hardware Specification

Arduino Pro is a microcontroller board based on ATmega168/328. In experiments, USB powered Arduino Pro (328) motes are used with the following configurations: Operating voltage - 3.3V, Clock speed - 8MHz, RAM - 2KB, FLASH - 32KB, EEPROM - 1KB, Radio module - nrf24L01, and Data rate - 19.2 Kbps.

Mica2 is a low-power sensor mote based on ATmega128L processor. USB powered Mica2 motes are used in the experiments with the following configurations: Operating voltage - 3.3V, Clock speed - 8MHz, RAM - 4KB, Flash - 128KB, EEPROM - 512KB, Radio module - CC1000, and Data rate - 19.2 Kbps.

5.4.1.2 Software Specification

The source code of each block cipher is written using Arduino IDE to compile and upload it on Arduino Pro motes. In our experiments, two built-in library functions ( microsecondsToClockCycles(), and Serial.print() ) are used to obtain and print information regarding the CPU cycles and encryption time.

A high-level component-based programming language (nesC) is used to implement the ciphers on Mica2 motes. The LocalTime.get() and printf() functions are
used to get the execution time, whereas the CPU cycles are obtained by using the
ATEMU.

Finally, the avr-size and avr-objdump utilities are used to measure the memory
usage on Arduino Pro and Mica2 motes respectively. These two utilities display the
header information of object files. The information includes the size of RAM and
ROM in terms of the text, data, and bss section.

5.4.2 Performance Evaluation and Analysis

This Section presents a comparative performance analysis of optimised Skipjack,
XXTEA, RC5, AES, and CGEA block ciphers implemented on Mica2 and Arduino
Pro motes. To make the comparison, three crucial parameters have been selected:
memory consumption, computational cost, and operation time. Figure 5.7 shows
the experimental setup for measuring the performance of block ciphers.

![Experimental setup: (a) USB powered Mica2 mote, (b) USB powered Arduino Pro with programmer board, (c) Battery powered Arduino Pro mote](image)

**Figure 5.7:** Experimental setup: (a) USB powered Mica2 mote, (b) USB powered Arduino Pro with programmer board, (c) Battery powered Arduino Pro mote
5.4.2.1 Memory Consumption

Memory consumption is a significant performance metric that can be used to select encryption algorithms with less memory overhead. Table 5.9 shows the amount of memory consumed by each block cipher on Mica2 and Arduino Pro platforms. It can be seen that Skipjack and AES-256 require more memory compared to other algorithms. The memory requirements of AES-128 is slightly lower compared to AES-256, whereas RC5 is the lightest among all algorithms.

Table 5.9: Memory consumption of block ciphers in bytes

<table>
<thead>
<tr>
<th>Block Ciphers</th>
<th>Mica2 RAM</th>
<th>Mica2 ROM</th>
<th>Arduino Pro RAM</th>
<th>Arduino Pro ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skipjack</td>
<td>3096</td>
<td>8658</td>
<td>398</td>
<td>4952</td>
</tr>
<tr>
<td>XXTEA</td>
<td>542</td>
<td>6312</td>
<td>226</td>
<td>4112</td>
</tr>
<tr>
<td>RC5</td>
<td>682</td>
<td>6110</td>
<td>350</td>
<td>3184</td>
</tr>
<tr>
<td>AES-128</td>
<td>1074</td>
<td>6296</td>
<td>814</td>
<td>3692</td>
</tr>
<tr>
<td>AES-256</td>
<td>1822</td>
<td>7932</td>
<td>1014</td>
<td>4190</td>
</tr>
<tr>
<td>CGEA</td>
<td>664</td>
<td>6268</td>
<td>548</td>
<td>3228</td>
</tr>
</tbody>
</table>

*Critical observations*- Both Skipjack and AES use a big S-box of 256-bytes, and as a result the algorithms occupy a significant amount of memory. XXTEA, RC5, and CGEA occupy very low memory, and thus they are suitable for memory constrained SNs like Arduino Pro. One important observation is that the implementation of AES-256 on Arduino Pro mote shows a message regarding low available memory. Therefore, cryptographic algorithms that use excessive memory may experience stability problems on Arduino Pro platform.

5.4.2.2 Computational Cost

The energy efficiency of an algorithm can be calculated from its computational complexity. Assuming the energy consumption per CPU cycle is fixed, the amount of consumed energy per byte can be computed by measuring the number of CPU
cycles required to process one byte of plaintext. However, Table 5.10 shows the total number of CPU cycles required by each algorithm to encrypt 32 bytes data. It can be seen that Skipjack is the most energy efficient block cipher, whereas the performance of AES-256 is worst among all algorithms. It is also noted that AES-128 performs two times better than AES-256.

Table 5.10: Computational cost of block ciphers in CPU cycles

<table>
<thead>
<tr>
<th>Block Ciphers</th>
<th>Mica2</th>
<th>Arduino Pro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skipjack</td>
<td>9820</td>
<td>12672</td>
</tr>
<tr>
<td>XXTEA</td>
<td>24064</td>
<td>30464</td>
</tr>
<tr>
<td>RC5</td>
<td>53014</td>
<td>61504</td>
</tr>
<tr>
<td>AES-128</td>
<td>37525</td>
<td>43200</td>
</tr>
<tr>
<td>AES-256</td>
<td>80344</td>
<td>88896</td>
</tr>
<tr>
<td>CGEA</td>
<td>67786</td>
<td>76212</td>
</tr>
</tbody>
</table>

**Critical Observations** - The key size and the number of rounds play a significant role in computational complexity. The implementation of AES-128 block cipher reduces more than half of computational cost required by AES-256 due to small size of key and fewer rounds. It is also noted that RC5 consumes more CPU cycles compared to AES-128 in spite of having the same size key. The reason is that RC5 executes 14 rounds, whereas AES-128 uses 10 rounds only.

### 5.4.2.3 Operation time

Operation speed or time efficiency is defined in terms of encryption time and communication time. Encryption time is the amount of time spent to encrypt the plaintext, whereas the time required to encrypt and successfully send the ciphertext is defined as communication time. Figure 5.8 shows the execution time required to encrypt 32 bytes data. It can be seen that Skipjack is more than 7 and 6 times faster compared to the AES and CGEA ciphers respectively. In addition, the AES-128 cipher reduces more than half of AES-256 encryption time. The same results are obtained for the
communication time experiment as shown in Figure 5.9.

![Figure 5.8: Elapsed time to encrypt 32 bytes data](image)

**Critical observations** - Skipjack algorithm is most efficient since it generates the shortest expanded key among all block ciphers. Similarly, the use of 128-bits key in AES-128 shows better performance compared to the AES-256 cipher. XXTEA also requires low encryption time since the cipher is structured with simple XOR and shift operations. The longer word size (32-bit) leads to longer execution times for both key setup and encryption phases in RC5. The CGEA block cipher also takes a significant amount of encryption time to perform crossover operations by repeatedly swapping the values at different memory locations.

### 5.4.3 Discussion

According to the experimental results, RC5 is the most memory efficient block cipher. XXTEA and CGEA are also potential candidates for memory constrained SNs like Arduino Pro. On the other hand, Skipjack shows the best performance in terms of the operation time and computational cost. XXTEA and AES-128 ciphers also
consume low energy. However, from a security perspective, Skipjack is a high risk algorithm because of a shorter key length. Similarly, XXTEA and RC5 are vulnerable to a number of security attacks such as timing attack, and chosen plaintext attack. Moreover, a 128-bits key is not secure against the quantum attack. The quantum computing systems are able to break a 128-bits key with time $2^{64}$ [167]. However, AES-256 and CGEA would still be secure against the exhaustive search attack due to their 256-bits key length. Therefore, we recommend to use the AES-256 or CGEA block ciphers when security is a priority. The RC5, XXTEA, and AES-128 ciphers can be used for the applications that require a minimum level security.

### 5.5 Summary

This chapter proposes two lightweight encryption algorithms which are suitable for a large volume of data encryption. SLES uses elliptic curve operations to generate key lists for each sensor node as well as in computing pseudorandom bit sequences for the encryption procedure. The algorithm uses an elliptic curve over prime field in order
to avoid high precision floating point calculations which are not suitable for sensor nodes. Similarly, the CGEA cipher uses the N-logistic tent map which deals with integer parameters and thus simplifies the computation process. Both algorithms use a common architecture, where the same parameters are used to perform many functions such as generation of the key list and pseudorandom bit sequences for the encryption procedure. Finally, the experiments on real platforms provide a good understanding of cost vs. performance relationship.

Although the proposed encryption schemes provide node verification and end-to-end data confidentiality, the route construction phase is still vulnerable to security attacks. The adversaries can capture the control messages transmitted during the route construction phase and use the information for malicious purposes. Furthermore, there should be a mechanism to defend against jamming attack and comprised node attacks in WSNs. To address the problems, the next chapter proposes an energy efficient secure multipath routing protocol which implements a number of security measures to build a secure communication channel for the entire communication process.
This chapter proposes a new Energy Efficient Secure Multipath (EESM) routing protocol which implements an energy balanced routing technique to prolong the network lifetime. A number of security measures such as OHC, MAC, key exchange mechanism, and encryption algorithm is tightly coupled with the proposed routing protocol to establish a secure channel between two communicating nodes as well as to minimise the implementation gap. Furthermore, a new node monitoring system is proposed to defend against the compromised node attack in EESM. The protocol is implemented on hardware to evaluate security overheads and setup time. It is also tested in a simulator, and the results are compared with the existing routing schemes.

6.1 Introduction

During the past few years, many single path and multipath routing protocols have been proposed for WSNs to extend the network lifetime. However, most of the protocols use shortest path algorithms to reduce the total energy dissipation. These energy optimising protocols find optimal routes by sacrificing uniform energy use in the network, and thus drain out energy of some of the SNs along those paths. Therefore, an energy balanced routing scheme is necessary to maintain a balance in energy consumption of the entire network.
Security is a critical concern for many WSN applications. Security operations-support multipath routing protocols provide both detection and prevention mechanisms against security attacks launched by the adversaries. However, designing secure multipath routing protocols requires special care since this can degrade the performance if not implemented in an efficient and effective way. The big challenge is to define the way through which the alternative paths are used in these protocols to maximise the security and reliability of transmitted data packets. A reputation-based trust scheme in multipath routing is a promising approach in WSNs. The advantage of this scheme is that it will add another layer of security to defend against malicious attacks. Thus, a reputation based trust management scheme is useful to maintain a trade-off among reliability, performance, and risks of node compromisation in the network.

This chapter proposes an energy efficient secure multipath routing protocol which securely constructs multiple routes for every SN so that the node can send data to the BS. The EESM protocol couples both routing and security mechanisms by combining the clique based clustering approach of the CBCR protocol, routing procedure of the IAHR protocol, key exchange mechanism of SLES, and encryption algorithm of the CGEA cipher. The proposed protocol integrates energy efficient routing with cryptographic mechanisms in order to provide a unique solution for WNS routing and security problems. The following are some unique benefits of the proposed protocol:

i) It implements an energy balanced route selection mechanism with the help of local and global knowledge of the network topology, and thus extends the network lifetime.

ii) The use of disjoint multiple routes increases network reliability since this minimises the damage caused by node failures and jamming attack. Furthermore, the use of location information of the BS and SNs makes it difficult to launch Sybil attacks and cloned node attacks in EESM.

iii) The proposed protocol implements a coordinator-based monitoring system which can effectively identify the compromised nodes in the network.
iv) The EESM protocol uses the same cryptographic scheme for MAC and OHC sequence generation, pairwise key distribution, and data encryption procedures in order to reduce the implementation gap.

6.2 Network Model and Assumptions

It is assumed that a WSN contains hundreds or thousands of randomly deployed SNs over the entire network area. All SNs are static and homogeneous in terms of communication range and initial energy. Every SN has a unique node ID and the SN is aware of the the location of the BS as well as its own. We assume that each SN has a communication range $d$ and the communication channels are symmetric, i.e., if a node $u$ can receive a message from $v$, it can also send a message to $v$. Unlike the resource limited SNs, the BS is sufficiently powerful to defend itself against security attacks. Furthermore, it is capable of computing routing information of every SN as well as maintaining the whole network. Finally, we assume that an adversary is computationally bounded and can pose the following security threats [67].

- An adversary has a jamming range $r$ such that $r \geq d$. Within the circle of radius $r$, the adversary can block or interfere other SNs by generating radio signals. However, the adversary is capable of jamming only a small part of the network i.e., $r < R$, where $R$ is the radius of the entire network.

- An adversary can physically capture a SN and obtain all of its secret information including cryptographic parameters and routing information. Furthermore, the adversary can also reprogram the compromised SN and turn it into a malicious node. However, it requires some time to compromise a node.

- An adversary can receive data from any SN or the BS within its communication range $d$. Thus, the packet acceptance range of the attacker is still $d$, whereas the jamming range may be greater than $d$.

We also assume that each node is preconfigured with a unique symmetric key, which is shared only with the BS. Each SN is also pre-distributed with a globally
known hash function \( F \) and an initial one-way hash chain number \( S_0 \). A temporary global MAC key is preconfigured to all SNs, which is used to protect the integrity of the broadcast messages. Furthermore, a seed key i.e., a distinct elliptic curve point is stored in each SN, which is used to generate a private key ring for that SN by using elliptic curve operations. For the sake of simplicity, the symmetric key can also be used as the seed key by a SN.

### 6.3 Energy Efficient Secure Multipath Protocol

The proposed protocol is divided into three parts: route discovery, data transmission, and network maintenance. Route discovery is for setting up the network topology by exchanging several control messages. It is subdivided into four phases. During the first phase, the BS securely floods a pairwise key setup (PKS) message in the network. Then every SN exchanges a pairwise key with its neighbour nodes and establishes a secure communication channel. In the second phase, a neighbour collection (NBC) message is broadcast in order to collect the neighbour information of each SN. Upon receiving the NBC message, the SNs securely send node IDs and locations of their neighbours to the BS using neighbour collection reply (NCR) messages. In the third phase, the BS divides the SNs into a number of groups and selects a coordinator for each group. The coordinators collect node information from the member nodes and forward them to the BS. Finally, the BS constructs multiple routes for every SN, and securely unicasts the routing table to the corresponding node. The data transmission phase is started as soon as the route discovery phase has been completed. In this phase, the SNs forward data packets to the BS using multiple routes. Network maintenance is the last phase of the EESM protocol which deals with the joining and leaving nodes, route updates, and node failures in the network. In the following, we present a detailed description for each phase.
6.3.1 Route Discovery: Pairwise Key Setup

Although the route discovery takes place after the deployment of sensors, it can also be initiated whenever the BS requires to construct the routing tables of all SNs. In this phase, the BS first broadcasts a pairwise key setup message to the entire network. The format of the PKS message is as follows:

\[
BS \rightarrow \ast: \text{PKS} || \text{OHC} || BS_{ID} || BS_{LOC} || \text{MAC}(K_{GM}, \text{PKS} || \text{OHC} || BS_{ID} || BS_{LOC}).
\]

where, PKS is the message type, BS_{ID} and BS_{LOC} are the ID and location of the BS respectively, OHC is a one-way hash chain sequence number, K_{GM} is the global MAC key, and || denotes concatenation.

Upon receiving the message, a SN first verifies the integrity of the message and rebroadcasts it in a modified format. As for example, a node V includes its own node ID and location in the rebroadcast message as well as replaces the previous MAC with the new computed MAC. Furthermore, the node V maintains a neighbour list and keeps records of the nodes from which it receives the broadcast messages. If the node receives another PKS message with the same OHC value, it only includes the ID and location in the neighbour list, but does not rebroadcast the message. The format of the PKS message sent by node V is as follows:

\[
V \rightarrow \ast: \text{PKS} || \text{OHC} || BS_{ID} || BS_{LOC} || V_{ID} || V_{LOC} || \text{MAC}(K_{GM}, \text{PKS} || \text{OHC} || BS_{ID} || BS_{LOC} || V_{ID} || V_{LOC}).
\]

After rebroadcasting the PKS message, each SN waits for a fixed period of time before starting the key exchange mechanism. In this stage, every SN exchanges a secret key with each of its neighbour nodes using a secure key pre-distribution scheme as described in Section 5.2.1. As soon as the SN completes setting up pairwise key with each of its neighbours, the node immediately erases the temporary global MAC key (K_{GM}). However, before erasing the key, every node generates a local MAC key for each of its neighbour node as well as its own. As an example, if node V has a
neighbour node $U$, then the local MAC key of node $U$ is computed by node $V$ such that $K_{Lu} = \text{MAC} \left( K_{GM}, U_{ID} \right)$. Now, node $V$ can authenticate the local broadcast message sent by node $U$ in the next phase using the key $K_{Lu}$.

A number of attacks can be launched in this phase including spoofing attack, DoS attack, Sybil attack, and rushing attack. The aim of spoofing attack is to impersonate the BS and direct all reply messages from the SNs to the attacker itself. Thus, the attacker can retrieve the topology information from captured messages and prevent the BS from receiving that information. Furthermore, the adversary can launch a DoS attack by flooding the entire network. The EESM protocol defends against all these attacks effectively with the help of OHC, MAC, and location information of the SNs and the BS. In the proposed protocol, the use of one-way hash chain value limits the capability of an adversary to flood the BS’s broadcast messages. The BS employs a one-way function $F$ to generate a sequence of values $S_0$, $S_1$, $S_2$, ..., $S_n$ such that $S_i = F(S_{i+1})$, where $0 \leq i < n$. Every node can verify the origin of the message by using its preconfigured hash function $F$ and the initial value $S_0$. Upon receiving the first broadcast message, a SN authenticates the originator of the message by verifying $S_0 = F(S_1)$. The one-way property of hash function $F$ ensures that only the BS can generate the next OHC number correctly. Thus, the mechanism prevents an adversary from spoofing the BS and injecting a forged PKS message. Furthermore, it is not possible to launch a rushing attack by replaying the fake PKS message. In EESM protocol, the recipient node always verifies the integrity of the received messages. Since every message includes the MAC of the immediate sender, it is not possible to inject spurious information unless the global MAC key is revealed to the adversary. However, an adversary still can capture the PKS message and replays it to the distant area of the network. But, the receiver node can easily detect the fake PKS message by checking the location of the BS and the distance between the BS and the SN. In addition to preventing rushing attack, the location information makes physical compromise of SNs very hard because two conflicting location claims by the same node can be identified by verifying their node positions [168].
The overall impact of these security measures is that the capability of a malicious node is confined into the jamming attack and dropping a pairwise key setup message. The effectiveness of message dropping and jamming attack is limited due to the flooding mechanism employed by the broadcast messages. To launch a successful attack, a malicious node has to compromise a SN and reveal the global MAC key. However, it requires sufficient time to capture a node, whereas the temporary global MAC key is destroyed as soon as the PKS broadcast message is expired.

6.3.2 Route Discovery: Neighbour Information Collection

After sending the PKS message in the first phase, the BS waits for a certain period of time so that all SNs be able to exchange the pairwise keys among themselves. Then, the BS broadcasts the NBC message to gather the neighbourhood information of all SNs in the network. Since every SN knows the local broadcast key of its neighbour nodes, this key is used to preserve the integrity of the transmitted messages in this phase. The format of the NBC message sent by the BS is as follows:

\[ BS \rightarrow * : \text{NBC} || \text{OHC} || \text{BS}_{ID} || \text{MAC} (\text{K}_L^{BS}, \text{NBC} || \text{OHC} || \text{BS}_{ID}). \]

When a node \( V \) receives this NBC message, it verifies the OHC value and MAC. If the OHC is valid, it stores the new OHC value by replacing the old one. Then, the message is rebroadcast in a modified format. The receiver node includes its node ID and replaces the old MAC with its new computed MAC. The format of the NBC message sent by node \( V \) is as follows:

\[ V \rightarrow * : \text{NBC} || \text{OHC} || \text{BS}_{ID} || \text{V}_{ID} || \text{MAC} (\text{K}_L^{V}, \text{NBC} || \text{OHC} || \text{BS}_{ID} || \text{V}_{ID}). \]

After forwarding the NBC message, every SN waits for some fixed period of time before sending the feedback message to the BS. In this stage, a node \( V \) selects the closest node to the BS from its neighbour list and unicasts the reply message. Since every SN is aware of the locations of its neighbours and the BS, it can easily compute the Euclidean distance between a neighbour node and the BS. If the distance is
same for more than one neighbours, the node with lowest node ID is selected to forward the reply message. The reply message of a node \( U \) contains the encrypted neighbourhood information of node \( U \) and is protected by a keyed MAC. The format of the NCR message sent by node \( U \) to its parent node \( V \) is as follows:

\[
U \rightarrow V : \text{NCR} \ || \ U_{ID} \ || \ E(K_{EU}, NI_U) \ || \ MAC(K_{PV}, \text{NCR} \ || \ \text{OHC} \ || \ U_{ID} \ || \ E(K_{EU}, NI_U)).
\]

where, NCR is the message type, \( NI_U \) and \( K_{EU} \) denote neighbourhood information and secret key of node \( U \) respectively, and \( K_{PV} \) is the pairwise key shared between node \( U \) and \( V \). On receiving the message, node \( V \) verifies the OHC and MAC first and then includes its node ID in the message. After that, it selects the closest neighbour to the BS as described earlier and computes a new MAC. The old MAC is replaced by the new MAC before rebroadcasting the message. If another SN receives the message, it replaces the node ID and MAC of \( V \) by its own and repeats the same procedure.

It is noted that the reply message uses an OHC number and the MAC. The OHC value prevents repetition of the same MAC in later rounds, whereas the MAC protects the reply message from being tampered. It also helps the destination node from receiving multiple copies of the same reply message. Furthermore, the confidentiality of the reply message is preserved against eavesdropping since the neighbour information is encrypted using the secret key of the source node.

### 6.3.3 Route Discovery: Coordinator Selection and Status Collection

In this phase, the BS first decrypts the neighbourhood information field of each NCR message and constructs a connectivity matrix. The matrix is a two dimensional array, in which, the connectivity between two nodes is presented by 1, and 0 represents no connectivity. The next step is dividing the SNs in a number of groups or clusters and then selecting a coordinator for each cluster. To form clusters, the BS
implements a clique-based clustering approach as described in the CBCR protocol. Although the maximal clustering approach implements the cluster formation process by the node with maximum connectivity, we have conducted experiments for both maximum and minimum degree nodes as shown in Figure 6.1. It can be seen that minimum degree clustering has fewer single node clusters compared to maximum degree approach. Since the aim of coordinator selection process is to keep all SNs under surveillance, cluster formation process started by the minimum degree node is preferred for the proposed routing scheme.

![Figure 6.1: Single node clusters](image)

After completion of the group formation process, the BS arbitrarily selects a coordinator for each group since all SNs in a group have almost the same amount of remaining energy at this stage. Then, the BS unicasts a group information (GIF) message to each group coordinator as shown below:

\[ BS \rightarrow V : \text{GIF} \parallel BS_{ID} \parallel D_{ID} \parallel D_{LOC} \parallel E(K_{E_{BS}} , GI_{D}) \parallel \text{MAC}(K_{P_{BS,V}} , \text{GIF}) \parallel BS_{ID} \parallel D_{ID} \parallel D_{LOC} \parallel E(K_{E_{BS}} , GI_{D}) \parallel \]
where, GIF is the message type, $D_{ID}$ and $D_{LOC}$ denote the node ID and location of the destination node, $GI_D$ is the group information encrypted by the private key shared between the BS and the destination node. The location information of the destination node is used by the intermediate node to select the next hop on the basis of Euclidean distance as mentioned earlier. When a node $U$ receives the message, it verifies the message integrity and finds out the next closest neighbour to the destination node. Then, it includes its own node ID and replaces the old MAC with the new MAC before forwarding the GIF message.

The coordinator nodes decrypt the GIF messages and retrieve group information that contains the node IDs of all group members and a group key. Every coordinator node sends this information to the group members by encrypting the information with corresponding pairwise key. After that, it broadcasts a request message to collect the status of the member nodes. This broadcast message contains a keyed MAC generated by the group key. When the member nodes receive the message, they send their current status to the coordinator. This status information includes the remaining energy and the link quality of each member node and is encrypted with a pairwise key. After receiving the status information from all member nodes, the coordinator node aggregates the information and unicasts them to the BS.

In addition to exchanging the group information, a coordinator node monitors the activities of its member nodes. If any suspicious activity such as dropping packets is noticed, it reports the IDs of the malicious nodes to the BS. Similarly, the member nodes can also report against the coordinator node if such behaviour is observed. Thus, the proposed routing scheme provides an efficient mechanism to detect the compromised nodes in the network.

### 6.3.4 Route Discovery: Route Selection and Distribution

In this phase, the BS constructs multiple routing paths for every SN using a heuristic function and a heuristic search algorithm as described in the IAHR protocol. The heuristic function combines the remaining energy, the link quality, and the local and global distance of a SN in order to derive the heuristic value of that node. This value
is used in the $A^*$ search algorithm to determine efficient routes between the source and the destination node. Suppose, the BS computes multipath forwarding tables to reach node $X$. First, it implements the $A^*$ search algorithm and finds a path to reach node $X$. After that, it excludes the nodes of the current path and constructs an alternative route to reach the destination. It may happen that no alternative path is found in some network scenarios. In such situations, the single path is used for data transmission. Figure 6.2 shows the multiple routes selection process executed by the BS.

The BS computes the forwarding table for each node and unicasts the table to the corresponding SN. To preserve confidentiality and integrity, the information is encrypted and MAC is added to the message.

### 6.3.5 Data Transmission

The sensor network starts its designated tasks in this phase. Now, every SN maintains a forwarding table with multiple entries, where each entry is defined by a 3-tuple: $(destination, source, intermediate node)$. For example, given a route from $S$ to the BS: $S \rightarrow i \rightarrow j \rightarrow k \rightarrow BS$, the forwarding table of node $i, j$ and $k$ will...
contain $|BS, S, S|$, $|BS, S, i|$, and $|BS, S, j|$ entries respectively. Thus, when a SN receives a data packet, it searches the forwarding table for an exact match. If it finds the entry in the table, it forwards the data packet.

The proposed protocol sets up an individual routing path for every SN in the network. Although this strategy ensures end-to-end confidentiality, it consumes a significant amount of energy. For many WSN applications, only the aggregator nodes send data packets to the BS in order to conserve energy. The EESM protocol can also provide data aggregation facility with the help of coordinator nodes. Instead of sending individual data packets to the BS, the SNs can forward their data packets to the coordinator nodes. The coordinators then aggregate and send the information to the BS.

### 6.3.6 Network Maintenance

The network maintenance phase deals with message loss, nodes joining and leaving, and route updates. A message can be lost due to collision, node failure, and compromised node attack. Since the broadcast messages are not acknowledged, it is not possible to identify message loss for those messages. Therefore, if a node has not received the PKS or NBC message, then it will not be able to continue the rest of the steps. Thus, the node will be disconnected from the network. However, the EESM protocol overcomes this problem using a local repair method. If a node $X$ does not receive either the PKS or NBC message for some time interval, it sends a joining request to its neighbour nodes. The format of the joining request (JRQ) message is as follows:

$$BS \rightarrow X: \text{JRQ} \| X_{ID} \| \text{ENC}(K_{EX}, (\text{JRQ} \| X_{ID})).$$

where, JRQ is the message type, $X_{ID}$ is the node ID of $X$, and $K_{EX}$ is the secret key shared with the BS. When the other nodes receive this message, they send it to the coordinator node. The coordinator forwards the message to the BS listing all node IDs from which it received the message. The BS verifies the integrity and
authenticity of the message. If the verification is successful, node $X$ is asked to exchange pairwise key with all its neighbours. Then, the BS sends the forwarding table and the current OHC value to the new node. However, it may be necessary that the groups have to be reformed in order to maintain complete connectivity among the group members. Figure 6.3 shows an example of such situation.

![Figure 6.3: The node joining process: (a) a network with 9 SNs and a BS, (b) the joining request of node $X$ is forwarded to the BS, (c) the BS reforms corresponding groups after verification](image)

One limitation of this approach is that the JRQ message is not verified by the neighbour nodes. Therefore, an adversary can launch a battery-drain attack by continuously sending ‘joining requests’. However, since limited number of nodes are joined after topology construction, this attack can be forestalled by setting a threshold level on joining requests. For example, a node is allowed to receive at best $n$ JRQ messages at each round. Thus, the damage caused by the power-drain or DoS attacks can be limited during the node joining phase.

Updating routes is another critical task which has to be maintained properly in order to keep the network functional. The BS periodically collects node status and keeps records of the energy consumption and link quality of the SNs. For example, the coordinator nodes consume more energy compared to the member nodes since it aggregates, sends and receives control messages. Thus, the coordinators will quickly
deplete their energy. To maintain an equal balance in energy consumption, the BS selects new coordinator for each group as necessary. Similarly, some nodes may be unavailable due to power failure or security attack. To overcome such conditions, the BS can update the current routes or even re-initiate the routing process.

6.4 Lifetime Analysis of WSNs

Network lifetime is a critical concern in WSN routing protocols design. Experiments and simulations are two ways that can be used to evaluate the lifetime of a WSN. However, these techniques have some limitations. Experiments with sensors may not be viable for the implementation in real scenarios. Furthermore, these manual experiments are costly and also need a long time to get data. On the other hand, simulation software has many inherent limitations regarding the hardware characteristics (e.g., buffer capacity) and communication technology capabilities (e.g., interference patterns). These shortcomings can be solved by analytical modelling, which defines the network lifetime through mathematical expressions.

This section presents an analytical model to formally define the lifetime of WSNs by considering a number of input factors including the remaining energy, the link quality, and the location of the sensor nodes in the network. The model derives an expression for lifetime estimation on the basis of the distribution of sensor nodes in the deployment area that can be used to evaluate the performance of any routing protocol. Furthermore, it is validated by simulations and has been found that the analytical outcomes are close to experimental results. Finally, the lifetime of a WSN is estimated using the proposed analytical model for both single path and multipath routing protocols.

6.4.1 Modelling Lifetime

The Lifetime of a Sensor Network (LSN) is defined as the duration (number of rounds) after which the network is disconnected due to the failure of one or more
SNs. However, this definition is application-specific. A network can become non-functional when a SN or a certain percentage of SNs exhaust the battery power. The basic assumptions of the proposed model are: i) the BS and SNs are stationary, ii) all SNs are homogeneous in terms of their capacity such as communication range and initial energy, iii) every SN is aware of the location of the BS as well as its own, iv) SNs employ a multi-hop routing scheme to reach the BS if the BS is out of their communication range. Furthermore, we consider the network area as a collection of unit rectangles, where every SN in a rectangle has equal probability of being selected as a forwarding node. The benefit of using a grid structure is that the entire network can be divided into sub-networks on the basis of multiple BSs and their area of coverage. Therefore, we can individually measure the lifetime of each sub-network and combine the results to obtain the total network lifetime. For the proposed model, the remaining lifetime of individual sensor is derived first.

6.4.1.1 Remaining Lifetime of an Individual Sensor

A sensor node consumes energy when it receives, transmits or listens to the wireless channel. For the sake of simplicity, we only consider energy dissipation involved in sending, receiving, and retransmitting messages. The Remaining Lifetime of an Individual Sensor (RLIS) can be defined as the ratio of the remaining energy to the initial energy. Thus, Remaining lifetime = (Initial energy - Consumed energy) / Initial energy. It can be expressed as follows according to the Sha-Shi model [169]:

\[
L(s) = 1 - \frac{1}{E_s} \sum_{i=1}^{N_m} \epsilon_{s_{iq}} N_{s_{iq}} R_t + \epsilon_{s_{ir}} N_{s_{ir}} R_t
\]  

(6.1)

where, \( L(s) \) is the remaining lifetime of sensor \( s \); \( N_m \) is the total number of queries or events; \( \epsilon_{s_{iq}} \) and \( \epsilon_{s_{ir}} \) are the amount of energy required to transmit and receive a query, and a reply message respectively; \( N_{s_{iq}} \) and \( N_{s_{ir}} \) are the number of transmitted and received query, and reply messages respectively by sensor \( s \) at moment \( i \) based on the probability \( P_s \); \( R_t \) is the expected number of retransmissions under the assumption that the query and reply messages experience the same link quality, and
Chapter 6. *Energy Efficient Secure Multipath Routing*

$E_s$ is the initial energy of node $s$. Now, based on the Heinzelman et al. model [43], we derive the radio hardware energy consumption for transmitting a $k$-bit message over distance $d$ (i.e., $E_{Tx}$) and receiving a $k$-bit message (i.e., $E_{Rx}$) as follows:

$$E_{Tx}(k, d) = k \epsilon_{elec} + k \epsilon_{amp} d^2$$

(6.2)

$$E_{Rx}(k) = k \epsilon_{elec}$$

(6.3)

where, $\epsilon_{elec}$ is the electronics energy to power the radio circuitry, and $\epsilon_{amp}$ is the amplifier energy for radio signal propagation for one bit information. Thus, the energy consumption can be expressed as: $\epsilon_{siq} = \epsilon_{siq,E_{Tx}} + \epsilon_{siq,E_{Rx}}$ and $\epsilon_{sir} = \epsilon_{sir,E_{Tx}} + \epsilon_{sir,E_{Rx}}$; where, $\epsilon_{siq,E_{Tx}}$ and $\epsilon_{sir,E_{Tx}}$ represent energy consumption for sending query and reply messages respectively, $\epsilon_{siq,E_{Rx}}$ and $\epsilon_{sir,E_{Rx}}$ denote energy consumption for receiving query and reply messages respectively.

Now, the probability of a message passing through sensor $s$ at moment $i$ ($P_{si}$) has to be calculated to compute the value of $N_{siq}$ and $N_{sir}$. Since the proposed model is based on the distribution of the SNs over the entire network, the deployment area is divided into a number of unit rectangles both horizontally and vertically as shown in Figure 6.4. The basic assumption is that the messages will pass through either vertical or horizontal unit rectangles depending on the locations of SNs and the BS. As an example, if the locations of the BS and the source node are (0, 0) and (0, 70) respectively, the messages will pass through horizontal unit rectangles to reach the BS. On the other hand, for a SN with (70, 0) location, the messages will go across vertical unit rectangles. Furthermore, since all SNs in unit rectangle have equal probability, the probability of passing a message through a SN within a rectangle is $\frac{1}{v}$, if there are $v$ sensors in that rectangle. Let $A$ and $B$ be two events, where $A$ denotes the occurrence of a message going through sensor $s$ at moment $i$, and $B$ denotes the event that the destination node is far away from node $s$ at moment $i$. Since these two events are independent, the probability of a message passing through sensor $s$ at $i$-th moment can be defined as follows:

$$P_{si} = P(A)_{si} P(B)_{si} = \frac{N - h \rho t}{Nh \rho}$$

(6.4)
where, $P(A)_{si} = \frac{1}{h\rho}$ and $P(B)_{si} = \frac{N-h\rho t}{N}$. The term $N$ denotes the total number of SNs, $h$ is the height of the rectangle, $\rho$ is the density of the SNs in the network, and $t$ is the number of unit rectangles between the sensor and the BS.

Finally, we calculate the packet retransmission rate based on the Automatic Repeat reQuest (ARQ) protocol. Let $P_e$ be the average packet error rate for the wireless channel, and $G$ be the maximum retransmission attempts allowed by the ARQ protocol. The packet error rate can be expressed as follows: $P_e = 1 - (1 - P_b)^m$; where $P_b$ is the bit error rate of the wireless channel, and $m$ is the number of bits in a packet [170]. If a packet is not successfully transmitted after $G$ attempts, then it is dropped. Hence, the expected number of retransmissions for a single packet can be defined as shown below:

$$R_t = \sum_{k=1}^{G} kF_t(k) = \sum_{k=1}^{G-1} [kP_e^{k-1}(1 - P_e)] + GP_eG = \frac{1 - P^G_e}{1 - P_e}$$  (6.5)
where, \( F_t(k) = \begin{cases} 
\frac{P_{e}^{k-1}(1 - P_{e})}{P_{e}^k} & 1 \leq k < G, \\
E_{s}(1 - P_{e}) & k = G
\end{cases} \)

By substituting the above values into Eq. (6.1), we can express the RLIS as follows:

\[
L(s) = 1 - \sum_{i=1}^{N_{m}} \frac{(1 - P_{e}^{G})(\epsilon_{sig}N_{sig} + \epsilon_{sir}N_{sir})}{E_{s}(1 - P_{e})}
\]  

(6.6)

### 6.4.1.2 Remaining Lifetime of a Sensor Network

The Remaining Lifetime of a Sensor Network (RLSN) can be obtained by aggregating the remaining lifetime of all individual sensors. However, the location of a SN in the network also has a significant impact on determining the network lifetime. For example, the SNs closer to the BS will quickly deplete their energy since they will forward more data packets than the SNs located far away from the BS. This means that the sensors closer to the BS should have more weights than the distant sensors. Therefore, the distance between the BS and a SN \( (d_{sb}) \) is inversely proportional to the weight factor \( (w_s) \) of that sensor. Thus, \( w_s = \frac{c}{d_{sb}} \), where \( c \) is a constant. Now, by combining the weight factor with the remaining lifetime of individual sensors, we can express the RLSN as follows:

\[
\Re = \sum_{s=1}^{N} \frac{c}{d_{sb}} L(s)
\]  

(6.7)

This equation can be used to define the lifetime of both single path and multipath routing protocols. As described earlier, analytical model is a more viable technique to measure the lifetime compared to experiments and simulations in many WSN applications. The difficulties involved in real implementation and the limitations of simulation platforms necessitate the design of analytical models for different network scenarios.
6.4.2 Lifetime Modelling: Single Path Protocol

In this section, the proposed model is used to determine the lifetime of a network under a conventional protocol such as a request and response protocol. In a request and response protocol, the SNs send request messages including their current node status (e.g., energy level, number of hops) to the neighbour nodes and the neighbour nodes select the best forwarding node on the basis of the received information. The query (request) message is sent to all SNs while the reply message (response) goes through only the sensors selected to establish the communication link. Therefore, using Eq. (6.4) and (6.6), we can define the RLIS for each query as follows:

\[
L(s) = 1 - \sum_{i=1}^{N_m} \epsilon_{siq} N_{ngh} (1 - P_e^G) \frac{E_s f (1 - P_e)}{E_s (1 - P_e)} + \frac{N \epsilon_{sir} P_{si} (1 - P_e^G)}{E_0 h \rho f (1 - P_e)}
\]

\[
= 1 - \frac{N m \epsilon_{siq} N_{ngh} (1 - P_e^G)}{E_0 (1 - P_e)} - \frac{N m \epsilon_{sir} (1 - P_e^G)(N - ht \rho)}{E_0 h \rho f (1 - P_e)} \tag{6.8}
\]

where, \(N_{ngh}\) is the number of neighbours of node \(s\), \(E_0\) and \(\frac{1}{f}\) are the initial energy and the probability of generating a response by a SN respectively. Now using Eq. (6.7) and (6.8), we can calculate the RLSN for the conventional protocol as follows:

\[
\mathcal{R} = \sum_{s=1}^{N} \frac{c}{d_{sb}} \left(1 - \frac{N m \epsilon_{siq} N_{ngh} (1 - P_e^G)}{E_0 (1 - P_e)} - \frac{N m \epsilon_{sir} (1 - P_e^G)(N - ht \rho)}{E_0 h \rho f (1 - P_e)} \right)
\]

\[
\approx c \ln(d_{max} + 1) (1 - \frac{N m \epsilon_{siq} N_{ngh} (1 - P_e^G)}{E_0 (1 - P_e)} - \frac{N m \epsilon_{sir} (1 - P_e^G)(N - ht \rho)}{E_0 h \rho f (1 - P_e)}) \tag{6.9}
\]

where, \(d_{max}\) denotes the maximum distance between the BS and a SN.
6.4.2.1 Verification and Analysis of the Model

The proposed model is evaluated both analytically and by simulation in order to validate its accuracy. For analytical investigation of RLIS and RLSN, we consider the following parameters. The initial energy of each SN is set to 1.725J. The energy consumption for sending and receiving a query message is 240µJ and 120µJ respectively, whereas the energy for transmitting and receiving a reply message is 1200µJ and 600µJ respectively [169]. The probability of packet error is set to 0.1, the maximum number of retransmissions is set to 3, and \( c \) is 0.01. We assume that 100 sensor nodes are randomly deployed in a 100 × 100 (m\(^2\)) area. Thus, the density of the SNs is 1 per 100 m\(^2\). Furthermore, we assume the average number of neighbour nodes for an individual sensor is 5 and the probability of generating a response by a sensor is \( \frac{1}{30} \), i.e., \( f = 30 \).

Now, using Eq. (6.8), we compute the remaining lifetime of different sensors located at different unit rectangles as shown in Table 6.1. It can be seen that the remaining lifetime of the sensors is increased when the distance between sensors and the BS is increased. This is because the SNs close to the BS consume more energy as they transmit and receive more data packets than the distant SNs.

<table>
<thead>
<tr>
<th>Unit Rectangles (t)</th>
<th>RLIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1 - ( \frac{4316N_m}{10^6} )</td>
</tr>
<tr>
<td>40</td>
<td>1 - ( \frac{3263N_m}{10^6} )</td>
</tr>
<tr>
<td>70</td>
<td>1 - ( \frac{2210N_m}{10^6} )</td>
</tr>
<tr>
<td>100</td>
<td>1 - ( \frac{1158N_m}{10^6} )</td>
</tr>
</tbody>
</table>

The remaining lifetime of the network is calculated using Eq. (6.9). Table 6.2 shows the RLSN values obtained for both single and multiple BSs. It can be seen that the RLSN is extended when the BS is located at the centre (in the case of a single BS scenario). As an example, if we set \( N_m = 250 \), the amount of remaining
energy is 0.035 and 0.012 for location of BS at (0, 0) and (50, 50) respectively. This remaining energy is wasted since at that time most of the nodes around the sink are out of power according to Table 6.1. However, the values indicate that the amount of unused energy is low when the BS is located at the centre. On the other hand, the RLSN is increased significantly in the case of multiple BSs scenario. In this scenario, the network becomes non-functional when the RLSN value is 0.019 for $N_m = 780$. Since the packets travel a relatively shorter distance, the networks sustains for more rounds compared to the single BS scenario.

<table>
<thead>
<tr>
<th>Number of BS</th>
<th>Location of BS</th>
<th>RLSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single BS</td>
<td>[0, 0]</td>
<td>$49487-N_m/10^6$</td>
</tr>
<tr>
<td></td>
<td>[50, 50]</td>
<td>$4262-124N_m/10^6$</td>
</tr>
<tr>
<td>Multiple BS</td>
<td>[25, 25], [100, 0]</td>
<td>$71279-67N_m/10^6$</td>
</tr>
</tbody>
</table>

### 6.4.2.2 Simulation Results

We implement the proposed model in MATLAB to verify the results of the analytical model. For each scenario, 100 nodes are randomly deployed in a $100 \times 100$ ($m^2$) area and simulations are run for 1000 trials to obtain the average performance measures. Table 6.3 shows the simulation parameters, which are commonly used in WSN experiments. Initial energy is set to 1.75 joules for all SNs, whereas the transmission range is limited to 10 metres. The size of broadcast packets and data packets is set to 30 bytes and 150 bytes respectively. Finally, energy consumption by transmitter electronics and amplifier characteristic constant is set to 50nJ/bit and 10pJ/bit/m² respectively.

First, we calculate the remaining lifetime of the SNs located at different rectangles after running the simulations for 700 rounds. In each round, every SN sends a data packet to the BS. Figure 6.5 shows that the nodes located near the BS possess very
Chapter 6. *Energy Efficient Secure Multipath Routing*

Table 6.3: Simulation parameters

<table>
<thead>
<tr>
<th>Types</th>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>Initial Energy</td>
<td>1.725 J</td>
</tr>
<tr>
<td>Application</td>
<td>Transmission Range</td>
<td>10 m</td>
</tr>
<tr>
<td></td>
<td>Packet Size</td>
<td>30, 150 bytes</td>
</tr>
<tr>
<td>Radio Model</td>
<td>$\epsilon_{\text{elec}}$</td>
<td>50 nJ/bit</td>
</tr>
<tr>
<td></td>
<td>$\epsilon_{\text{amp}}$</td>
<td>10 pJ/bit/m$^2$</td>
</tr>
</tbody>
</table>

The remaining lifetime of SNs increases with the increase of the distance (unit rectangles). However, the increase in remaining lifetime is not linear for all SNs at different locations. The reason is that the SNs near the centre of the network area consume more energy compared to the SNs close to the boundary due to an increase in the network connectivity and collisions. The analytical results also indicate that the closer the SN is to the BS, the lower the energy it would have. Thus, the simulation results confirm the validity.
of the analytical model.

Figure 6.6 illustrates the simulation results of the remaining energy of the network after 700 rounds. Remaining energy is the total amount of energy that the network possesses after sending and receiving data packets for a number of rounds. It can be seen that the remaining energy of BS at (0, 0) location is lower than that of BS at (50, 50) until three nodes run out of energy. Since packets travel a long distance to reach the BS, a significant amount of energy is consumed in receiving and forwarding data packets. However, after that the remaining energy for BS at (0, 0) location increases and the trend is continued for rest of the dead nodes. When more nodes exhaust their battery, the network is partitioned and many SNs cannot communicate with the BS in spite of having a sufficient amount of energy. In contrast, SNs remain alive for more rounds and consume more energy if the BS is positioned at the centre. Therefore, the topology with the BS at centre makes better use of the node energy compared to the (0, 0) location in the long run. Both simulation and analytical results (Table 6.2) show that deploying the BS at the centre significantly extends network lifetime.

![Remaining Energy Graph](image)

**Figure 6.6:** Remaining energy of the network
Figure 6.7 shows the total number of rounds until the network becomes non-functional. The network is considered non-functional when 10% of the total nodes is dead. From the graph, it can be seen that the network sustains for 868 rounds for the centre location, whereas it remains alive for 741 and 706 rounds for the corner positions. The BS located at the corner or edge is surrounded by fewer SNs compared to the BS at the centre. Thus, the network goes down when the surrounding nodes run out of power. This leads to the network partition and wastes a significant amount of energy. However, both the remaining energy and the network lifetime analysis validate the results shown in Table 6.2.

![Figure 6.7: Network lifetime for different BS locations](image)

### 6.4.3 Lifetime Modelling: Multipath Protocol

Multipath routing protocol uses more than one route to transmit data packets from a source node to a destination node. As for example, every SN sends data packets using two node-disjoint routes in the EESM protocol. Therefore, every SN consumes energy for either sending its own data packets or forwarding data packets of the other
nodes. Thus, the energy consumption of a SN $s$ at moment $i$ can be expressed as $\epsilon_{sid} = \epsilon_{sid.E_{Tx}}$ and $\epsilon_{sir} = \epsilon_{sir.E_{Tx}} + \epsilon_{sir.E_{Rx}}$, where $\epsilon_{sid.E_{Tx}}$ presents energy consumption for sending own data packets; $\epsilon_{sir.E_{Tx}}$ and $\epsilon_{sir.E_{Rx}}$ denote energy dissipation for receiving and forwarding data packets from other SNs respectively. Let, every SN sends two data packets ($N_{sid}$) through different routes in each round. Then, the RLIS of node $s$ in EESM protocol can be defined as follows:

$$L(s) = 1 - \left( \frac{\epsilon_{sid}N_{sid}R_t + \sum_{j=1}^{N-1} \epsilon_{sir}N_{sir}R_t}{E_s} \right)$$

$$= 1 - \left( \frac{2\epsilon_{sid}(1-P^G_e) + \sum_{j=1}^{N-1} \epsilon_{sir}(N-h\rho t)(1-P^G_e)N_{sir}R_t}{E_s(1-P_e)} \right)$$

$$= 1 - \left( \frac{2\epsilon_{sid}(1-P^G_e)}{E_s(1-P_e)} + \frac{(N-1)(N-h\rho t)\epsilon_{sir}(1-P^G_e)}{2E_s h\rho f(1-P_e)} \right)$$

(6.10)

where, number of forwarded packets, $N_{sir} = P_{si} \times \frac{1}{f}$, $\frac{1}{f}$ denotes the probability of forwarding data packets by a SN, and $E_s$ is the initial energy of node $s$.

Substituting the value of RLIS in Eq. (6.9), the RLSN can be computed as follows:

$$\Re = \sum_{s=1}^{N} \frac{c}{d_{sb}} \left( 1 - \left( \frac{1-P^G_e}{E_s(1-P_e)} \left( 2\epsilon_{sid} + \frac{(N-1)(N-h\rho t)\epsilon_{sir}}{2h\rho f} \right) \right) \right)$$

$$\approx c \ln(d_{max} + 1)(1 - \left( \frac{2(1-P^G_e)}{E_0(1-P_e)} \left( \epsilon_{sid} + \frac{(N-1)(N-h\rho t)\epsilon_{sir}}{4h\rho f} \right) \right)$$

(6.11)

where, $d_{max}$ denotes the maximum distance between the BS and a SN, and $E_0$ is initial energy of a SN. Now, using Eq. (6.11), the ratio of the remaining energy per round can be determined.

The following parameters are considered to compute the RLSN in EESM protocol: $E_0 = 2J$, $\epsilon_{sid} = 1200\mu J$, $\epsilon_{sir} = 1800\mu J$, $P_e = 0.1$, $G = 3$, $f = 50$, $h = 100$, $\rho = 0.01$, $t = 50$, $d_{max} = 71$, $N = 100$, and $c = 0.2$. Thus, the remaining lifetime of a sensor network with 100 nodes is 0.81. Similarly, the RLSN for 200, 300, 400, 500 nodes are 0.78, 0.76, 0.74, and 0.73 respectively. Notice that an increase in the number of nodes results an increase in node density, packet forwarding probability of a SN, and collisions in a fixed size network.
6.5 Implementation of EESM Protocol

The EESM protocol was implemented on a network of ten Arduino Pro sensor motes equipped with nrf24 radio modules. A BS was implemented using a USB powered Arduino board connected to a laptop. All compute-intensive processes were performed by the BS to conserve SN’s energy. Furthermore, the reply messages are segmented into multiple packets, if the size is more than 32 bytes. Every segment is assigned a distinct sequence number so that any tampering with the sequence number or contents of a segment packet can be detected by the BS.

6.5.1 Security Measures

A number of different security measures is implemented in the EESM protocol namely the one-way hash chain, keyed MAC, data encryption, and key exchange mechanism. All these mechanisms are implemented through a common security framework that uses almost the same security parameters and structure to perform different tasks.

Every SN is preconfigured with a unique symmetric key or seed key shared with the BS. This key is a distinct point in an elliptic curve which can generate a number of random points (i.e, key pool) using point addition and point doubling elliptic curve operations. The seed key is used with other parameters to encrypt data packets between a SN and the BS, whereas the pairwise key (an elliptic curve point in the key pool) is used for encrypting information between two nodes. Notice that the values of $x_i$ and $y_i$ are updated after each iteration to avoid repetition in the bit sequence. However, the key exchange mechanism described in sub-section 5.2.1 and the encryption algorithm presented in section 5.3 are used in the EESM protocol.

Finally, the same encryption algorithm, CGEA is used in generating keyed MAC and one-way hash chain numbers. A standard cipher block chaining mode was used to generate each MAC as shown in Figure 6.8. To generate a one-way hash chain, the CGEA cipher is implemented in such a way that the secret key cannot be revealed.
for a given plaintext and the corresponding ciphertext. Figure 6.9 shows the one-way hash chain generation process run by the BS. First, the BS chooses a random key $S_n$ and uses it to encrypt a known plaintext. This results in ciphertext $S_{n-1}$, which is used as key to encrypt the same plaintext to generate $S_{n-2}$. This process is continued to compute the complete sequence of the one-way hash chain, i.e., $S_{n-1}, S_{n-2}, \ldots, S_1, S_0$.

![Figure 6.8: CBC-based message authentication code procedure](image)

![Figure 6.9: One-way hash chain generation procedure](image)

### 6.5.2 Security Overhead Analysis

To evaluate the computational cost and memory usage of different cryptographic schemes, we implemented the one-way hash chain verification, encryption, and decryption algorithms on Arduino Pro motes by using the RC5 and CGEA ciphers. RC5 is a flexible block cipher with variable parameters, whereas CGEA is a lightweight block cipher based on the pseudorandom bit sequences. The 256-bit block of the random bit sequences is used as the key to encrypt or decrypt 128-bit
data blocks. The algorithm implements XOR, mutation, and crossover operations on the plaintext to generate the ciphertext. Instead of using rounds, it performs crossover operation for each byte of data in the plaintext. Here, we implemented 14-rounds RC5 algorithm with a 128-bit key, and 64-bit data blocks. Table 6.4 shows a comparative view of elapsed time in millisecond (ms) and memory usage in bytes by these two algorithms for 32 bytes data.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Procedure</th>
<th>Time (ms)</th>
<th>RAM (bytes)</th>
<th>ROM (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC5</td>
<td>Encryption</td>
<td>7.26</td>
<td>350</td>
<td>3184</td>
</tr>
<tr>
<td></td>
<td>Decryption</td>
<td>8.09</td>
<td>350</td>
<td>3496</td>
</tr>
<tr>
<td></td>
<td>OHC</td>
<td>14.16</td>
<td>436</td>
<td>3890</td>
</tr>
<tr>
<td>CGEA</td>
<td>Encryption</td>
<td>9.28</td>
<td>548</td>
<td>3228</td>
</tr>
<tr>
<td></td>
<td>Decryption</td>
<td>9.68</td>
<td>548</td>
<td>3228</td>
</tr>
<tr>
<td></td>
<td>OHC</td>
<td>17.73</td>
<td>692</td>
<td>4164</td>
</tr>
</tbody>
</table>

It can be seen that the RC5-based cryptographic schemes perform better than the CGEA-based schemes. Because, CGEA requires a significant amount of time to perform crossover operations which iteratively swap values at different memory locations. Furthermore, RC5 uses a 128-bit key, whereas CGEA generates 256-bit pseudorandom bit sequences for its encryption/decryption procedure. Another important observation is that the verification of one-way hash chain requires more time compared to the encryption and decryption procedures.

### 6.5.3 Network Setup Time

Network setup time is the total time required to make the network operational. The time interval between sending the first broadcast message and receiving acknowledgements of routing tables delivery from all SNs by the BS is defined as network setup time. There are a number of activities that affect the setup time such as execution of cryptographic schemes, packet processing or data aggregation, sending and
receiving control messages, waiting time before starting the next step, and random delay. In this experiment, the BS waits for 500 ms after receiving a reply message and the time is reset after every new reply message. The BS starts computing routing tables after either receiving reply messages from all SNs or detecting a time out event. Similarly, the waiting time is set to 500 ms for every SN before starting the pairwise key exchange with its neighbours and sending neighbour information to the BS. In contrast, the waiting time is set to 100 ms for unicast messages sent by the BS.

Figure 6.10 shows the total network setup time for different network size. It can be seen that the implementation of RC5-based cryptographic schemes requires less setup time compared to the CGEA-based implementations. Although the RC5-based algorithms perform better than the CGEA-based schemes, CGEA is preferred for an enhanced level of security. Furthermore, CGEA is suitable for designing a common security architecture which is essential to minimise security threats in WSNs.

![Figure 6.10: Network setup time for different network size](image_url)
6.5.4 Throughput and PDR Analysis

The two of the most important performance metrics throughput and Packet Delivery Ratio (PDR) are used to evaluate the proposed routing protocol in a real environment. Average throughput is measured by calculating the total number of bytes received by the BS per unit time. On the other hand, PDR is obtained by measuring the ratio of number of packets received at the BS to the number of packets sent by the sensor nodes. In this experiment, the EESM protocol is implemented for two different scenarios: i) normal indoor setup, and ii) presence of couple of interfering devices. According to vendors, the communication range of Arduino Pro motes with 250kbps data rate is 100 meters for outdoor setup. However, in our indoor setup, the sensors were unable to communicate with each other if they were kept more than about 40 meters apart. To observe the impacts of level of interference, a microwave oven and a wireless router are placed at two different locations: 10 and 20 meters away from the sensor motes and the BS. Both interfering devices operate at 2.4 GHz radio frequency and thus they affect data transmission in the network. The experiment is run for 200 seconds, where every sensor mote sends a 32 bytes data packet to the BS after every 50 milliseconds. To support long distance transmission, data rate is set to 250kbps for every sensor mote. Further, auto acknowledgement is turned off so that the sensors need not to wait for acknowledgements before transmitting next data packets. Similarly, auto retransmit is explicitly disabled to keep track of the number of dropped packets in the network. Table 6.5 shows the experimental outcomes of real implementation of the EESM protocol for both single path and multipath routing.

It can be seen that both single path and multipath routing achieve a good level of PDR and almost the same average throughput when there is no additional source of interference. However, the performance degrades with the presence of external sources of interference. In this situations, multipath routing achieves higher levels of PDR and average throughput compared to the single path routing. Furthermore, performance of single path routing is highly affected when the interfering devices are located closer, at 10 meters distance to the sensors. In this case, multipath
Table 6.5: Throughput and PDR Analysis

<table>
<thead>
<tr>
<th>EESM Protocol</th>
<th>Location of Interfering devices (meter)</th>
<th>Average Throughput (kbps)</th>
<th>PDR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Interference</td>
<td>50.03</td>
<td>98.68</td>
<td></td>
</tr>
<tr>
<td>Single Path</td>
<td>20</td>
<td>47.88</td>
<td>91.54</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>41.67</td>
<td>71.23</td>
</tr>
<tr>
<td>No Interference</td>
<td>51.07</td>
<td>99.26</td>
<td></td>
</tr>
<tr>
<td>Multipath</td>
<td>20</td>
<td>49.85</td>
<td>97.84</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>46.79</td>
<td>89.43</td>
</tr>
</tbody>
</table>

Routing achieves more than 5% average throughput and 18% PDR gain compared to single path routing. Since the sensor motes use the predefined routes, some nodes are unable to send data packets to the BS due to high level of interference. On the other hand, the multipath routing overcomes the problem by using alternative routes and thus obtains a high level of PDR and average throughput.

6.6 Resilience to Security Attacks

The proposed EESM protocol is resilient against several attacks such as Sybil attack, rushing attack, DoS attack, jamming attack, compromised nodes attack, wormhole attack, and sinkhole attack. In a wormhole attack, two malicious nodes use a private, out-of-band channel and then encapsulate the routing paths of their neighbour nodes. On the other hand, in a sinkhole attack, a malicious node advertises fake information to attract the SNs. However, this attack is not effective for a centralised routing protocol, where only the BS computes the routing tables. In our proposed routing scheme, the BS constructs routing paths for each SN and these paths are updated periodically. Furthermore, the integrity and/or confidentiality is preserved for every message and therefore, a malicious node is unable to attract a SN using fake information. Thus, the proposed protocol effectively defends against both wormhole
and sinkhole attacks. This section describes the resiliency of the EESM protocol against node failures, jamming attack, and compromised node attack.

### 6.6.1 Node Failure

In sensor networks, node failure is a common scenario which mainly happens due to power dissipation or node compromisation. However, the effects of node failure can be estimated by modelling the probability of path failures in the network.

Let $S$ be a set of $n$ disjoint paths such as $S = \{P_1, P_2, \ldots, P_n\}$, where $P$ denotes a path between the source and the destination nodes. Let $F_{P_i}(\Delta t)$ be the probability of failure of a path $P_i$ among those $n$-paths in a time interval $\Delta t$. Further, we assume that individual path failure is independent. It means that the failure of a path will not affect the other path of the same node since the paths are completely node-disjoint. Thus, the probability of failure of all $n$-paths in the time interval $\Delta t$ can be defined as follows:

$$F_{P_\forall}(\Delta t) = \prod_{i=1}^{n} F_{P_i}(\Delta t) \quad (6.12)$$

Since the EESM protocol does not support node mobility, the path failures only occur due to node failures. Hence, the probability of an individual path failure can be defined in terms of availability of the nodes in that corresponding path as shown below:

$$F_{P_i}(\Delta t) = 1 - \prod_{j \in P_i} A_j(\Delta t) \quad (6.13)$$

where, $A_j(\Delta t)$ denotes the availability of node $j$ on path $P_i$ during time $\Delta t$. Now, substituting the value of individual path failure in Eq. (6.12), the probability of failures of all paths can be determined as per the following equation:

$$F_{P_\forall}(\Delta t) = \prod_{i=1}^{n} \left(1 - \prod_{j \in P_i} A_j(\Delta t)\right) \quad (6.14)$$

The above equation states that the possibility of failures of multiple paths can be determined using the knowledge of node availability of each path. The increase in
the probability of failure of multiple paths will definitely result in an increase in the number of affected nodes.

Figure 6.11 shows the average number of blocked nodes with respect to different number of failed nodes. In this experiment, we simulated both single-path and multipath versions of the proposed protocol, where the communication range was set to 15m and 30m respectively. The measurements were performed for a network of 500 randomly deployed SNs over a 100 × 100 m² area. Furthermore, the simulation was performed for 500 times to compute the average value. The result indicates that the multipath scheme employed by the EESM protocol significantly increases the robustness of the network over the single-path approach. It is also noted that the number of blocked nodes decreases with the increase of communication range. The reason is that a SN can directly communicate with more nodes due to an increase in network connectivity.

![Figure 6.11: Effects of node failures](image-url)
6.6.2 Jamming Attack

The reliability of wireless communications is affected by the jamming attack since it can disrupt the communication between two nodes. A jamming node can block all the nodes within its communication range and this range is usually lower than the radius of the entire network ($R$). Thus, a jamming node can affect only limited number of nodes depending on its jamming range. Given a WSN consists on $N$ nodes distributed on a two-dimensional plane over an area of side length $L \times L$. Let a malicious node $M$ has a jamming range $r$ such that $r \geq d$. Then, the jamming capability of $M$ is confined into a small part of the network, i.e., $r < R$. A jamming node can be positioned either inside or outside of the network. However, if it is inside the network then the average degree of node affected by the jamming node is:

$$M_{avg} = \rho \pi r^2 = \frac{N \pi r^2}{L^2}$$  \hspace{1cm} (6.15)

where, $\rho$ is node density in the network. The number of affected node increases with the increase of jamming range. As an example, if $N = 100$, $L = 100$, and $r = 20$, then $M_{avg} \approx 13$, whereas $M_{avg} \approx 51$, when $r = 40$. However, since jamming range is lower than the network radius, the number of affected nodes, $M_{avg} < N$.

Figure 6.12 shows the resilience of the EESM protocol against the jamming attack by assessing the average number of blocked nodes as a function of the number of jamming nodes. In this experiment, the network configuration is same as the previous one except that the communication range of SNs is set to 15m, whereas the range is set to 15m and 30m respectively for the jamming nodes. It can be seen that the adversaries can block a huge number of nodes as the jamming range increases. However, for both scenarios, the multipath routing performs better than the single-path routing since the connectivity between SNs and the BS is increased in a multipath routing scheme. Thus, the resilience of the network against the jamming attack is improved significantly.
6.6.3 Compromised Node Attack

A compromised node attack is a powerful attack in which an adversary steals the secret keying materials from a compromised node and uses the information to perform malicious activities. Since compromised nodes behave like the legitimate nodes in the network, it is difficult to detect those misbehaving nodes. However, in the EESM protocol, if a member node does not forward data packets or sends the packets to a malicious node, the coordinator can detect easily by using its message overhearing ability. Similarly, if the coordinator node misbehaves, it can also be identified by the member nodes. Furthermore, if a compromised node sends fabricated messages to the BS, the BS can detect the inconsistencies comparing them with the earlier messages.

Figure 6.13 shows the average number of affected nodes as a function of number of compromised nodes in the network. The same network configuration is used in this experiment as described for the node failure scenario. However, ten to fifty compromised nodes are randomly chosen out of 500 nodes. These compromised nodes are able to perform the following malicious activities: dropping messages,
fabricating messages, and forwarding messages to some unknown nodes. The results indicate that fewer SNs are affected in multipath routing compared to the single path routing protocol. The reason is that SNs can avoid the routes including compromised nodes if alternative path is available. However, this is not possible in single-path routing and thus, large number of nodes are unable to send their data packets to the BS.

Further, the detection capability of the compromised node attack is presented in Figure 6.14. It can be seen that the detection rate is high when few nodes are compromised. With the increase of compromised nodes in the network, the detection rate degrades. If more nodes are compromised then the nodes can communicate with each other and mislead the BS by providing false information. An an example, let a group have five nodes. The coordinator and two other nodes are compromised by an adversary. Now, these compromised nodes can report the BS claiming the two benign nodes as compromised nodes. When the BS will get alert messages from all five nodes, it will definitely trust the claim of compromised nodes. However, this situation can be avoided by slightly modifying the current reporting system.
Instead of group monitoring, every node will be monitored by all its neighbours. If any inconsistency is found, it will be notified to the BS and then the BS will take the decision on the basis of majority. Thus, the detection process will be distributed rather than centralised one.

6.7 Performance Analysis

The proposed EESM protocol is implemented on the OMNeT++ simulator and compared with the SEEM and SRMR protocols. SEEM is a secure and energy efficient routing, whereas the SRMR protocol provides secure and reliable data delivery in WSNs. Three performance metrics: network lifetime, packet delivery ratio (PDR), and control overhead are chosen to evaluate the efficiency of the protocols. Two malicious nodes are randomly deployed in the network. One node is used to launch security attacks by taking multiple identities, whereas the other node generates random levels of interference varying its transmission range.

The simulation parameters used in the experiments are listed in Table 6.6. The base station is placed at the centre (50, 50) of a $100 \times 100$ m$^2$ area and it has infinite power supplies. On the other hand, the initial energy of each sensor is set to 2 joules and the transmission range is set to 15 metres for each protocol. Furthermore, the size of control packets is 24 bytes long, whereas the size of data packets is 128 bytes. Finally, energy consumption by the transmitter electronics and the amplifier characteristic constant is set to 50nJ/bit and 10pJ/bit/m$^2$ respectively.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Grid</td>
<td>(0, 0) to (100, 100)</td>
</tr>
<tr>
<td>Location of BS</td>
<td>(50, 50)</td>
</tr>
<tr>
<td>Initial Energy</td>
<td>2 J</td>
</tr>
<tr>
<td>Transmission Range</td>
<td>15 m</td>
</tr>
<tr>
<td>Broadcast Packet Size</td>
<td>24 bytes</td>
</tr>
<tr>
<td>Data Packet Size</td>
<td>128 bytes</td>
</tr>
<tr>
<td>$\epsilon_{elec}$</td>
<td>50 nJ/bit</td>
</tr>
<tr>
<td>$\epsilon_{fs}$</td>
<td>10 pJ/bit/m$^2$</td>
</tr>
</tbody>
</table>

6.7.1 Network Lifetime

The network lifetime is defined as total number of rounds that the network remains functional. The increase in the network lifetime indicates the better use of the node energy. Figure 6.15 shows the network lifetime comparison of the EESM, SEEM, and SRMR protocols before the network becomes non-functional. The network is considered non-functional when 10% of the total nodes run out of power. It can be seen that the EESM protocol achieves 25-37% increase in the network lifetime over the SEEM and SRMR protocols for densely deployed networks. The reason is that the EESM protocol implements energy balanced routing mechanism, whereas SRMR
protocol has to send redundant information for each data packet. On the other hand, SEEM consumes a significant amount of energy to broadcast the neighbour collection reply messages since it includes node IDs of all the neighbours for each node. Furthermore, a malicious node can launch a battery-drain attack in SEEM by persistently sending broadcast messages with different node IDs. Similarly, SRMR protocol is vulnerable to the HELLO flood attack that exhausts node energy fast.

![Figure 6.15: Comparison on network lifetime](image)

### 6.7.2 Packet Delivery Ratio

Packet delivery ratio is defined as the ratio of the number of received data packets to the number of transmitted data packets. Figure 6.16 shows the packet delivery ratios for different number of SNs. It can be seen that the EESM protocol achieves a higher level of PDR compared to the SEEM and SRMR protocols. The EESM routing algorithm considers link quality as one of the routing metric. Thus, the protocol can avoid the jammed nodes when selecting multiple routes. Furthermore, nodes are alive for more rounds in the EESM protocol compared to the other two
protocols and therefore, most of the links are available to forward data packets. This results in increased volume of data delivery in the proposed scheme. It is also noted that the PDR is increased with an increase in the number of nodes in the network. Since the network area is fixed, more data packets are delivered when the network connectivity is increased.

![Comparison on packet delivery ratio](image)

**Figure 6.16:** Comparison on packet delivery ratio

### 6.7.3 Control Overhead

Control overhead is defined as the ratio of the total number of control messages exchanged between the source and the destination nodes, and the total number of data packets that are successfully received at the destination node. To ensure the efficient use of network resources, the control overhead should be low. Figure 6.17 shows the control overhead of the EESM, SEEM, and SRMR protocols. Although the total number of control messages exchanged in the proposed protocol is slightly higher than that of the SEEM protocol, successful packet reception rate of EESM is significantly greater than that of the SEEM protocol. On the other hand, the
SRMR protocol has to send 6 redundant packets for every 15 data packets and thus incurs significant amount of redundancy (i.e., 40%). Therefore, EESM outperforms the other two protocols in terms of control overheads.

![Graph comparing control overhead](image)

**Figure 6.17:** Comparison on control overhead

### 6.8 Summary

The key objectives of the EESM protocol are to securely construct routing paths and transmit data packets in addition to network lifetime maximisation. To ensure the authenticity, integrity, and confidentiality, EESM employed a common security framework based on a robust cryptographic scheme, CGEA. This common framework generates a distinct elliptic curve point for each SN and this point is used as a secret key in the encryption/decryption procedure, and as a seed key for generating the private key pool. The one-way hash chain and MAC are also implemented using the CGEA cipher to tightly couple all these security measures. Furthermore, a new group monitoring system is proposed to effectively detect the compromised node attack. In addition to addressing all these security aspects, the
design of the EESM protocol also moves computation-intensive operations away from the resource-constrained SNs towards the resource-rich BS. Simulation results indicate that the proposed scheme effectively defends against several malicious attacks launched by the adversaries. Further, performance evaluation results show that EESM routing scheme outperforms the SEEM and SRMR protocol in terms of network lifetime, PDR, and control overheads.

To secure route construction phase, EESM implements several security mechanisms and these mechanisms can also be implemented in single path routing protocols. Thus, single path routing protocols can effectively defend against a number of security attacks like multipath routing protocols. However, the use of a single predefined path is a major limitation of the single path routing, and thus the protocols are still vulnerable to node failures, jamming attack, and compromised node attack. Therefore, multipath routing is more suitable than the single path routing for the applications which require an enhanced level of security and reliability. Implementation on real sensors also show that multipath routing outperforms single path routing in terms of PDR and average throughput.
This chapter summarises the research contributions provided by this thesis. In summary, the contributions of this dissertation are as follows: i) proposing two energy efficient single path routing protocols, ii) developing two lightweight cryptographic schemes to provide secure key establishment, node verification, and end-to-end data confidentiality, iii) designing a multipath routing protocol which integrates the proposed routing and security schemes, and iv) deriving an analytical model for lifetime estimation of WSNs. Finally, some open research challenges are highlighted to provide directions for potential future works.

7.1 Research Contributions

Limited functional capabilities, power and environmental factors, topology management complexity, and bandwidth-limited transmission channel poses several challenges in WSNs. These constraints make data aggregation, routing, and network maintenance in WSNs more difficult compared to wired networks. Security and energy efficiency are two primary concerns in the design of WSN routing protocols since WSNs are typically deployed in hostile environments and 70% of the total energy is consumed for data transmission. Based on the design constraints and research objectives, the work described in this dissertation has proposed a number of energy efficient routing protocols and lightweight security algorithms. Furthermore,
the proposed security mechanisms are combined with a routing protocol to establish a secure channel for the entire communication.

The contributions explained in Chapter 4 tested Hypotheses 1 and 2, which also helped to answer RQ-1 and RQ-2 respectively. This chapter presented the design and implementation of two energy efficient single path routing protocols in WSNs. Since WSN covers a wide range of application domains, it is essential to develop application specific protocols in addition to the general purpose routing protocols. A hierarchical clique-based clustering and routing protocol, and an interference aware heuristic routing protocol are designed for large-scale sensor networks and small-scale networks respectively with the following features to meet the research goals.

**Maximum network lifetime, increased PDR, and minimum latency:** Since sensor nodes are battery powered, reduction of energy consumption extends the network lifetime. Instead of the global clique formation process, the CBCR protocol forms local cliques to reduce the energy dissipation. Furthermore, the use of the TDMA scheme allows nodes to remain in the sleep state for a large amount of operation time. In this case, sensor nodes only need to awake during their specific slot to transmit data and thus conserves a significant amount of energy. Furthermore, CBCR minimises energy consumption through aggregation of correlated data. Since CBCR is a cluster-based protocol, it performs local data aggregation on the correlated data and can greatly reduce energy dissipation. Unlike the CBCR protocol, IAHR uses local and global topology information to find an energy efficient route to reach the destination node. The protocol uses an energy balanced routing technique so that critical nodes remain alive for a long period of time as well as the network remains functional. In addition, the protocol quantises the level of interference and uses it as one of the routing parameters. Thus, it takes every opportunity to transmit data using low-interfered links and reduces energy consumption required to determine the new route and send data through it.

PDR is defined as the ratio of the number of received data packets to the number of delivered data packets. Since nodes remain alive for a long period of time in the CBCR and IAHR protocols, they achieve a high level of PDR. The energy-balanced
routing helps to extend the lifetime of the SNs, and thus delay the network partitions. This results in increased level of data delivery since most of the nodes in the network are reachable. Similarly, the CBCR and IAHR protocols reduce average packet delay or latency. In the CBCR protocol, the nodes send their data packets to the cluster heads which are responsible to transmit them to the BS. Thus, the protocol minimises the route discovery time since only cluster heads construct effective routes to reach the destination node. Furthermore, data aggregation reduces the amount of transmitted data in the network and results in minimum latency of getting output to the end user. On the other hand, IAHR uses even low-interfered links to send data packets. Instead of marking a high-interfered link as unstable, it periodically checks the status of that link and starts to send data packets through the link when the interference value is below the threshold level. Simulation results show that the CBCR protocol achieves upto 40% increase in the network lifetime. Similarly, the IAHR protocol achieves 10-15% more PDR gain compared to the GAHR protocol in a noisy communication environment.

Chapter 5 proposed a key exchange mechanism and two lightweight encryption algorithms for resource limited sensor nodes. The proposed algorithms implement elliptic curve and genetic operations in order to design lightweight cryptographic schemes. The contributions explained in this chapter tested Hypothesis 3, which helps to answer RQ-3. It is required to implement different security measures in WSNs to improve the security of both single path and multipath routing protocols. However, the implementation of cryptographic schemes always incurs costs. Therefore, it is necessary to maintain a good trade-off between the cost and the performance. To address these issues, we have designed two robust and provably secure encryption algorithms with a key exchange mechanism as well as investigated the performance of a number of block ciphers in real sensor motes. The following are the key features of the proposed schemes which are implemented to meet the design goals.

**Fast, robust, and computationally secure:** Due to the limited computation and communication resources, the cryptographic schemes in WSNs should be simple and fast. The SLES and CGEA ciphers use an elliptic curve over prime field
and an N-logistic tent map in generation of the key pool and the pseudorandom bit sequences. Many encryption algorithms involve an LFSR or high-precision floating point calculations to compute the pseudorandom bit sequences, which consumes a significant amount of the CPU cycles. However, SLES and CGEA use only integer arithmetic and thus speed up the computation process. The experimental results (Table 5.2 and 5.6) show that both algorithms perform better than the other algorithms in terms of CPU cycles and elapsed time. Furthermore, the simple architecture of SLES and CGEA make them lightweight and suitable for sensor motes. It has also been shown that the proposed encryption algorithms are provably secure and robust against a number of security attacks. This is achieved by using the pseudorandom bit sequences as one time passwords in SLES. Since the nodes share a distinct elliptic curve point at the beginning of every new session, SLES generates completely different bit sequences each time for two communicating nodes. On the other hand, genetic operations over the plaintext based on the number of 1’s in the pseudorandom bit sequences are performed as tools for introducing diffusion and confusion properties in the CGEA cipher. The mutation and crossover operations create relatively fair diversity in the ciphertext. The concrete security evaluation and performance analysis show that the algorithms can resist all well known attacks.

Finally, a number of block ciphers including CGEA are implemented on Mica2 and Arduino Pro sensor motes to evaluate the performance in real environment. The results indicate that CGEA requires less memory, computation, and operation cost than the other 256-bit cipher, AES. From security perspective, CGEA is highly recommended since the key space of this algorithm can defend against powerful attacks such as quantum attack.

The contributions explained in Chapter 6 tested Hypotheses 3 and 4, which also helped to answer RQ-3a and RQ-4 respectively. This chapter proposes an energy efficient secure multipath routing protocol which integrates several security measures with the IAHR protocol to provide secure communication. To reduce the implementation gap among security algorithms and routing protocol, the EESM protocol is designed with the following features to meet the research goals.
Energy efficient, secure, use of common security framework: Multipath routing with security support is more reliable and secure than the single path routing due to use of redundant paths in data transmission. However, construction and maintenance of multiple routes requires exchange of control messages, which results in energy dissipation. Therefore, it is important to use energy efficient mechanisms in order to extend the network lifetime. To achieve energy efficiency, the EESM protocol implements the IAHR protocol to determine multiple node-disjoint paths. However, the main difference is that the route selection process is centralised rather than distributed since the base station constructs routes for each sensor node in EESM. Thus, the design of the EESM protocol moves all computation intensive operations towards the resource-rich base station and conserves energy of the sensor nodes. Experimental results in Section 6.7 indicate that the EESM protocol outperforms the SEEM and SRMR protocols in terms of the network lifetime, PDR, and control overheads. The EESM protocol improves the network lifetime by a maximum of 25% and 37% over the SEEM and SRMR protocols respectively.

The another key feature of the EESM protocol is secure construction of routing paths and data transmission. This is achieved by implementing a number of cryptographic schemes such as OHC, MAC, key exchange, and encryption mechanisms. All these mechanisms are implemented using the same security framework with the help of the CGEA cipher. In addition, a clique-based leader selection and group monitoring scheme is also introduced where every node in a group is under the surveillance of other member nodes for entire communication. This strategy effectively reduces the impacts of compromised nodes in the network. Furthermore, the use of MAC and OHC ensures the authenticity and integrity of both data packets and control messages. Apart of the experimental outcomes, an analytical expression is derived to estimate the lifetime of WSNs. Using the expression, the lifetime of a WSN for single path and multipath routing protocols is also predicted. The benefit of the analytical model is that it can be used as a tool for performance evaluation of any routing protocol.


7.2 Future Work

Energy efficiency and security are two key research areas in WSNs and there is still much work to be done. Since sensor networks are data centric and application specific, the design requirements may differ from application to application. Although the protocols are developed for both small-scale and large-scale WSNs in this research, there are many other WSN applications, where this is not the case. For example, mobility is an important routing attribute that has significant impacts on network performance. WSN applications such as industrial control, inventory management may use more mobile nodes than the static nodes. Since the proposed protocols support limited node mobility, they may not be applicable in those scenarios. Therefore, protocol architectures need to be developed to support mobility in greater extent. The mobility of SNs as well as the BS has to be considered while designing routing protocols for WSNs.

An elliptic curve with a 128-bits key is used to implement SLES since the successful attack against elliptic curve cryptography could break 112-bits key over prime field. However, the rapid advances in computational power and cloud computing make it easy to break 128-bit keys. As an example, a quantum computer requires between 1300 and 1600 qubits only to break a 224-bit elliptic curve cryptography. However, it is recommended to use an elliptic curve with 256-bits key to provide an enhanced level of security and this will definitely increase computational cost and memory usage. Therefore, it is necessary to investigate the performance of the proposed encryption algorithms for extended key length. Furthermore, the rapid growth rate of quantum computers also necessitates the development of post-quantum cryptography and DNA cryptography for secure communication in WSNs.

While we defined analytical expression for lifetime estimation, the energy consumption for sending, receiving, and retransmitting messages were considered for the sake of simplicity. However, energy use in listening communication channel, state transition, and data processing also has to be integrated to improve the accuracy of the proposed model. In addition, it is important to incorporate energy dissipation in the route construction phase in order to find how accurate the analytical model can
estimate the lifetime with respect to the simulation results. The analytical model for multipath routing also needs to be verified through simulation.

A compromised node detection mechanism is proposed in this dissertation based on misbehaviour detection. Every SN observes malicious behaviours of the neighbour nodes within the group and reports the events to the BS. The BS decides whether to isolate a node from the network on the basis of maximum number of votes. However, the proposed node monitoring system is a basic approach and does not consider all types of misbehaviours such as false praising, transient behaviour, and routing loop. The proposed scheme also need to be combined with the existing weighting approaches to develop a more effective trust based system.

Finally, the proposed single path routing protocols (i.e, CBCR, and IAHR) have been tested in simulators only and some basic assumptions are made to simplify the network scenarios. However, in real environment these assumptions may have significant impacts on the performance of the proposed schemes. Similarly, the network performance of the EESM protocol is measured using the Omnet++ simulator as well as on real sensors for a small network. Confirmation that these routing protocols can be verified in a real environment for large scale WSNs is still unknown. Therefore, the implementation of the proposed protocols for large networks has to be done to evaluate their effectiveness in real environments.
Appendix

The following simulation, emulation, and programming tools are used to implement and evaluate the routing protocols and encryption algorithms in this dissertation. Furthermore, the Mica2 and Arduino Pro sensor motes are used to measure the performance of the routing and cryptographic schemes in real environment.

A.1 Matlab

Matlab 7.14 is used to implement the CBCR protocol and to evaluate the security strengths of the CGEA cipher. The Matlab simulator provides a number of unique features which make it easy to gain insight into data in a fraction of time. Furthermore, it has many built-in functions and they are used to perform security analysis in our proposed protocols. For example, the `imread()` function is employed to read the gray-scale image and store the information in an $m \times n$ array. On the other hand, the `imwrite()` function is used to write image data from a two-dimensional array to its corresponding file. Similarly, the histograms of the plain and cipher images are obtained by using the built-in `histogram()` function, whereas the `corrcoef()` and `entropy()` functions are used to measure the correlation of the adjacent pixels and the entropy of the plain and cipher images respectively. The following Matlab code performs the operations described above.
A.2 OMNeT++

OMNeT++ is a discrete event simulator, is used to evaluate the performance of the IAHR and EESM protocols. Modular design, strong GUI support, and mobility framework are some unique features of OMNeT++, which provide an easy simulation platform for the users. These features allow for simulating open systems of wireless sensor nodes where both static and mobile nodes could be deployed to drive the simulations.

We used OMNeT++ (4.2.2 IDE) which includes the INET and mobility framework with a number of built-in network protocols such as AODV. The IDE supports codes written in the C++ language. In the experiments, all functional units are implemented in modules to ensure the reusability of the code. For example, node module is designed to handle the operations performed by a SN in the network. This module enables self-initialisation, error handling, message passing, data transmission, and reception in the network. The following sendDataNode2BS() function of node module sends data packet to the BS. The function first sets the fields of
the $DataNode2BSMsg$ message by assigning the message type, source address, destination address, and payload. Then, the built-in functions $findGate()$ and $gate()$ are used to check the existence of the specified outgate and connect it to the corresponding channel object. Finally, the OMNeT++ function, $isConnected()$, ensures the connectivity between in and out gates. If the connection is successful, the message is send to the BS and node energy is reduced on the basis of the packet size.

```c
void Node::sendDataNode2BS(int data)
{
    if (this->nodeStatus!=ALIVE)
        return;

    int rec = this->bsId;

    /* create a Node2BS message */
    DataNode2BSMsg *msg = new DataNode2BSMsg();
    msg->setType(ND_BSDATA);
    msg->setSrcAddress(this->nodeId);
    msg->setDestAddress(rec);
    msg->setData(this->nodeData);
    this->nodeData++;

    char gid[32];
    sprintf(gid, "0%ld", rec);
    if (findGate(gid) > 1)
    {
        cGate *g = gate(gid);
        if (g->isConnected())
        {
            send((DataNode2BSMsg *) msg, gid);
            /* reduce energy level */
            this->engTras(dlen * 8, this->bsDist);
        }
    }
}
```
A.3 TOSSIM

TOSSIM, a TinyOS simulator, is used to measure the memory consumption and operation time by the proposed encryption algorithms. The reason of using TOSSIM in our experiments is that it is a very powerful bit-level discrete event simulator. In TOSSIM, every node can be evaluated under perfect transmission conditions. Therefore, it provides more precise simulation results at component levels compared to the other low-level simulators.

TOSSIM supports nesC programs, which are built out of components. These components are connected and compiled to a single unit by the nesC compiler. A nesC program has two parts as shown in the following nesC code. The first, the encRC5 module, contains the executable logic of encRC5 which states that the module interacts with the rest of the system via two interfaces, Boot and Leds. The second part of encRC5, the encRC5AppC configuration establishes the connection with TinyOS’s services.

```nesC
module encRC5 { /* First part of nesC program */
    uses interface Boot;
    uses interface Leds;
}
implementation{
    uint32_t start_time, stop_time;
    event void Boot.booted() {
        start_time = call LocalTime.get();
        call Leds.led0Toggle();
        RC5_SETUP(); // Key setup procedure
        RC5_ENCRYPT(); // Encryption procedure
        call Leds.led0Toggle();
        stop_time = call LocalTime.get();
    }
}
```
A.4 Arduino IDE

Arduino IDE is an open source software platform used to write the code, compile and upload the program to the Arduino mote. The IDE supports C/C++ programming language including a heaps of library functions. In our experiments, `microsecondsToClockCycles()` and `Serial.println()` functions are used to obtain and print the CPU cycles and encryption time for the evaluated block ciphers.

```c
#include <stdlib.h>
#include <stdio.h>
#include "aes256.h"

unsigned long us, ms, ck;

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    aes256_context ct;
```
Appendix A. Appendix

```c
uint8_t key[33] = "HELLOALLHELLOALLHELLOALLHELLOALL";
uint8_t in[17] = "ABCDABCDABCDABCD";

aes256_init(&ct, key); // Key expansion procedure
aes256_encrypt(&ct, in); // Encryption procedure

us = micros(); // us since program start
ms = millis();
ck = microsecondsToClockCycles(us);

Serial.println(us, DEC);
Serial.println(ck, DEC);
```

A.5 ATEMU

ATEMU, a fine grained WSN emulator is used to calculate the elapsed CPU cycles by the encryption algorithms. Obtaining accurate CPU cycles from real sensors is a bit tricky since it requires special hardware setup. ATEMU can be used to solve this problem since the obtained CPU cycles by this emulator is very close to the measurements performed on real sensors. Therefore, we used the ATEMU emulator to compute the elapsed CPU cycles by the proposed encryption schemes in our experiments.

The AVR processor based emulation engine of ATEMU is employed to accurately model the execution of the code on each sensor by using the same binary image loaded onto the Mica2 node. The emulator includes a graphical interface Xatdb which provides debugging and monitoring facilities to the users. Using the Xatdb interface, we could record the CPU cycles when an event is triggered on/off. Figure A.1 shows the execution of an encryption algorithm in ATEMU.
A.6 Mica2

Mica2 is an ATmega128L microcontroller based sensor mote, which operates on the TinyOS. It can run sensor application and the radio communication stack simultaneously using TOS. To evaluate the performance of the SLES, AES, XXTEA, RC5, CGEA, and Skipjack ciphers, we used a USB-powered Mica2 sensor mote with CC1000 radio module. The following figure shows the MICA2 mote used in the experiments.
Appendix A. Appendix

Figure A.2: A Mica2 Mote

A.7 Arduino Pro

Arduino Pro is a low cost sensor mote based on ATmega168/328 microcontroller. It provides UART TTL serial communication and includes a serial monitor. This allows ASCII data to be sent to and from the Arduino board via a USB connection. We used Arduino Pro (328) sensors with nrf24L01 radio module to implement the AES, XXTEA, RC5, CGEA, and Skipjack ciphers. Figure A.3 shows the Arduino Pro mote used in performance evaluation of the block ciphers.

Figure A.3: An Arduino Pro Mote
A.8 Miscellaneous

Elliptic Curve Operations: Point addition and point doubling are two elliptic curve operations that can be used to generate random points on an elliptic curve. Let \( P(x_P, y_P) \) and \( Q(x_Q, y_Q) \) be two distinct points such that \( P(x_P, y_P) \) is not \(-Q(x_P, -y_P)\), then we could derive a new point \( R = P + Q \) through point addition operation using the following equations:

\[
S = \frac{y_P - y_Q}{x_P - x_Q}
\]

\[
x_R = S^2 - x_P - x_Q \quad \text{and} \quad y_R = -y_P + S(x_p - x_R)
\]

where, \( S \) is the slope of the line through the points \( P \) and \( Q \). Given an elliptic curve equation \( y^2 = x^3 - 7x \) and two points \( P(-2.35, -1.86) \) and \( Q(-0.1, 0.836) \) on the curve. Now, using the above equations, we can generate a new point \( P + Q = R = (3.89, -5.62) \).

![Figure A.4: Point addition](image)
Similarly, the point doubling operation can be used to obtain a new point on the curve. If \( y_P \) is not zero, then a new point \( R \) can be generated such that \( R = 2P \) using the following equations:

\[
S = \frac{3x_P^2 + A}{2y_P}
\]

\[
x_R = S^2 - 2x_P \quad \text{and} \quad y_R = -y_P + S(x_P - x_R)
\]

As an example, let \( P(2, 2.65) \) be a point on the elliptic curve \( y^2 = x^3 - 3x + 5 \).

Now, doubling the point \( P \), we can obtain a new point \( R(-1.11, 2.64) \), such that \( P + P = 2P = R \).

---

**Flowchart of Pseudorandom bit sequence generation in SLES:** In SLES, an elliptic curve point is shared between two nodes at the beginning of communication. This point is used as the base point to generate the pseudorandom bit sequences using point addition and point doubling operations as explained above. The following figure shows the steps performed in the pseudorandom bit sequences generation process.
Coefficients, $A, B$
Base Point, $G (x, y)$
Prime Number, $p$

Generate a new point $G' (x', y')$ using Point addition/doubling

$x' > y'$

Add $(x' \mod 2)$ in the bit sequence
Add $(y' \mod 2)$ in the bit sequence

Got desired length?

Yes

No

Yes

No

Figure A.6: The Pseudorandom bit sequences generation process
# List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>Advanced Encryption System</td>
</tr>
<tr>
<td>AODV</td>
<td>Adhoc On-Demand Vector</td>
</tr>
<tr>
<td>ATRM</td>
<td>Agent-based Trust and Reputation Management</td>
</tr>
<tr>
<td>ATSR</td>
<td>Ambient Trust Sensor Routing</td>
</tr>
<tr>
<td>BFS</td>
<td>Breadth First Search</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>CBCR</td>
<td>Clique Based Clustering and Routing</td>
</tr>
<tr>
<td>CGEA</td>
<td>Chaotic-map and Genetic-operations based Encryption Algorithm</td>
</tr>
<tr>
<td>CH</td>
<td>Cluster Head</td>
</tr>
<tr>
<td>CM</td>
<td>Connectivity Matrix</td>
</tr>
<tr>
<td>CMRP</td>
<td>Cluster based Multipath Routing Protocol</td>
</tr>
<tr>
<td>COA</td>
<td>Ciphertext Only Attack</td>
</tr>
<tr>
<td>CPA</td>
<td>Chosen Plaintext Attack</td>
</tr>
<tr>
<td>CRPV</td>
<td>Constant Random Perturbation Vector-based</td>
</tr>
<tr>
<td>CV</td>
<td>Coefficient of Variance</td>
</tr>
<tr>
<td>DARPA</td>
<td>Defence Advanced Research Project Agency</td>
</tr>
<tr>
<td>DHAC</td>
<td>Distributed Hierarchical Agglomerative Clustering</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial of Service</td>
</tr>
<tr>
<td>E2DSR</td>
<td>Energy Efficient Dynamic Source Routing</td>
</tr>
<tr>
<td>E-ADCR</td>
<td>Energy-efficient Asynchronous low Duty-Cycle Routing</td>
</tr>
<tr>
<td>EAMR</td>
<td>Energy-efficient ACO-based Multipath Routing</td>
</tr>
<tr>
<td>ECC</td>
<td>Elliptic Curve Cryptography</td>
</tr>
<tr>
<td>EECR</td>
<td>Energy Efficient Cluster-based Routing</td>
</tr>
<tr>
<td>EEGA</td>
<td>Energy Efficient Geocast Algorithm</td>
</tr>
<tr>
<td>EESM</td>
<td>Energy Efficient Secure Multipath</td>
</tr>
<tr>
<td>FH-CDMA</td>
<td>Frequency Hopping Code Division Multiple Access</td>
</tr>
<tr>
<td>FLOC</td>
<td>Fast Local Clustering Service</td>
</tr>
<tr>
<td>FND</td>
<td>First Node Dies</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GAEMW</td>
<td>Genetic Algorithm for Energy-entropy based Multipath routing in WSNs</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>GAF</td>
<td>Geographic Adaptive Fidelity</td>
</tr>
<tr>
<td>GAHR</td>
<td>Greedy and A* Heuristic Routing</td>
</tr>
<tr>
<td>GEAR</td>
<td>Geographic and Energy Aware Routing</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GPSR</td>
<td>Greedy Perimeter Stateless Routing</td>
</tr>
<tr>
<td>GBDD</td>
<td>Grid Based Data Dissemination</td>
</tr>
<tr>
<td>HCC</td>
<td>Hierarchical Control Cluster</td>
</tr>
<tr>
<td>HEED</td>
<td>Hybrid Energy-Efficient Distributed</td>
</tr>
<tr>
<td>HGR</td>
<td>Hybrid Geographic Routing</td>
</tr>
<tr>
<td>HIGHT</td>
<td>High Security and Lightweight</td>
</tr>
<tr>
<td>HIKE</td>
<td>Hierarchical Key Establishment</td>
</tr>
<tr>
<td>HND</td>
<td>Half of the Nodes Dies</td>
</tr>
<tr>
<td>IAHR</td>
<td>Interference Aware Heuristic Routing</td>
</tr>
<tr>
<td>IAODV</td>
<td>Improved Ad-hoc On-Demand Distance Vector</td>
</tr>
<tr>
<td>IND-CPA</td>
<td>INDistinguishability under a Chosen Plaintext Attack</td>
</tr>
<tr>
<td>INSENS</td>
<td>INtrusion-tolerant routing protocol for wireless SEnsor NetworkS</td>
</tr>
<tr>
<td>IV</td>
<td>Initialization Vector</td>
</tr>
<tr>
<td>JERT</td>
<td>Just Enough Redundancy Transmission</td>
</tr>
<tr>
<td>KPA</td>
<td>Known Plaintext Attack</td>
</tr>
<tr>
<td>LBRP</td>
<td>Location-Based Routing Protocol</td>
</tr>
<tr>
<td>LCA</td>
<td>Linked Cluster Algorithm</td>
</tr>
<tr>
<td>LEACH</td>
<td>Low Energy Adaptive Clustering Hierarchy</td>
</tr>
<tr>
<td>LF</td>
<td>Longevity Factor</td>
</tr>
<tr>
<td>LFSR</td>
<td>Linear Feedback Shift Register</td>
</tr>
<tr>
<td>LQ</td>
<td>Link Quality</td>
</tr>
<tr>
<td>MAC</td>
<td>Message Authentication Code</td>
</tr>
<tr>
<td>MDS</td>
<td>Maximum Distance Separable</td>
</tr>
<tr>
<td>MECH</td>
<td>Maximum Energy Cluster Head</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micro-Electronic-Mechanical-Systems</td>
</tr>
<tr>
<td>MERR</td>
<td>Minimum Energy Relay Routing</td>
</tr>
<tr>
<td>MFR</td>
<td>Most Forward within Radius</td>
</tr>
<tr>
<td>MND</td>
<td>Most of the Nodes Dies</td>
</tr>
<tr>
<td>MVMP</td>
<td>Multi-Version Multi-Path</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standard and Technology</td>
</tr>
<tr>
<td>NPCR</td>
<td>Number of Pixels Change Rate</td>
</tr>
<tr>
<td>NSA</td>
<td>National Security Agent</td>
</tr>
<tr>
<td>OHC</td>
<td>One-way Hash Chain</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>PEACH</td>
<td>Power Efficient and Adaptive Clustering Hierarchy</td>
</tr>
<tr>
<td>PEGASIS</td>
<td>Power Efficient Gathering in Sensor Information Systems</td>
</tr>
<tr>
<td>PIKE</td>
<td>Peer Intermediaries for Key Establishment</td>
</tr>
<tr>
<td>PRF</td>
<td>Pseudo-Random Function</td>
</tr>
<tr>
<td>PRSA</td>
<td>Path Redundancy based Security Algorithm</td>
</tr>
<tr>
<td>RDR</td>
<td>Randomized Dispersive Route</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RPB</td>
<td>Random Perturbation Based</td>
</tr>
<tr>
<td>ROL</td>
<td>Rotate Left</td>
</tr>
<tr>
<td>RQ</td>
<td>Research Question</td>
</tr>
<tr>
<td>RS</td>
<td>Reed-Solomon</td>
</tr>
<tr>
<td>RSS</td>
<td>Received Signal Strength</td>
</tr>
<tr>
<td>SDC</td>
<td>Secure Distributed Clustering</td>
</tr>
<tr>
<td>SEDR</td>
<td>Secure and Energy-efficient Disjoint Route</td>
</tr>
<tr>
<td>SEEM</td>
<td>Secure and Energy Efficient Multipath</td>
</tr>
<tr>
<td>SenIT</td>
<td>Sensor Information Technology</td>
</tr>
<tr>
<td>SHPER</td>
<td>Scaling Hierarchical Power Efficient Routing</td>
</tr>
<tr>
<td>SLES</td>
<td>Simple Lightweight Encryption Scheme</td>
</tr>
<tr>
<td>SN</td>
<td>Sensor Node</td>
</tr>
<tr>
<td>SPIN</td>
<td>Sensor Protocols for Information via Negotiation</td>
</tr>
<tr>
<td>SRMR</td>
<td>Secure and Reliable Multipath Routing</td>
</tr>
<tr>
<td>SWOT</td>
<td>Strength, Weakness, Opportunities, Threats</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>TEA</td>
<td>Tiny Encryption Algorithm</td>
</tr>
<tr>
<td>TEEN</td>
<td>Threshold Sensitive Energy Efficient Sensor Network protocol</td>
</tr>
<tr>
<td>TOSSIM</td>
<td>Tiny OS SIMulator</td>
</tr>
<tr>
<td>UACI</td>
<td>Unified Average Changing Intensity</td>
</tr>
<tr>
<td>WEP</td>
<td>Wired Equivalent Privacy</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
<tr>
<td>WHAN</td>
<td>Wireless Home Automation Network</td>
</tr>
</tbody>
</table>
Bibliography


