

AMERICAN UNIVERSITY OF BEIRUT

REPUTATION-BASED TRUST MODEL FOR SECURE
ROUTING IN WIRELESS SENSOR NETWORKS

by
AYMAN SOUHEIL TAJEDDINE

A dissertation
submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
to the Department of Electrical and Computer Engineering
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
February 2015

AMERICAN UNIVERSITY OF BEIRUT

REPUTATION-BASED TRUST MODEL FOR SECURE
ROUTING IN WIRELESS SENSOR NETWORKS

by
AYMAN SOUHEIL TAJEDDINE

Approved by:

Dr. Ayman Kayssi, Professor
Electrical and Computer Engineering


Chairman

Dr. Ali Chehab, Professor
Electrical and Computer Engineering


Advisor


Dr. Imad Elhadj, Associate Professor
Electrical and Computer Engineering


Member of Committee

Dr. Weisong Shi, Professor
Computer Science, Wayne State University


Member of Committee

Dr. Jean-Marc Seigneur, Senior Lecturer-Research Manager
Computer & Network Security, University of Geneva


Member of Committee

Dr. Wassim Itani, Assistant Professor
Electrical and Computer Engineering, Beirut Arab University


Member of Committee

Date of dissertation defense: February 3rd, 2015

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Tajeddine Ayman Souheil
Last First Middle

Master's Thesis Master's Project Doctoral Dissertation

I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

I authorize the American University of Beirut, **three years after the date of submitting my thesis, dissertation, or project**, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.



Signature

Feb. 19th, 2015

Date

ACKNOWLEDGMENTS

My sincerest gratitude and appreciation are addressed to my advisor, Prof. Ayman Kayssi for his continuous support, encouragement, and enlightening advice throughout this work. I appreciate all the time and effort he has put into this.

I also extend my appreciation to the insightful dissertation committee: Prof. Ali Chehab, Prof. Imad Elhajj, Prof. Weisong Shi, Prof. Jean-Marc Seigneur, and Prof. Wassim Itani. Thank you for your constructive comments and advice that greatly enhanced the quality of this work.

I also express my warmest gratitude to the love of my life, my wife Noura. Thank you for your continuous support and patience all through. I am so blessed to have you by my side.

Last but not least, I would like to express my deepest gratitude to my parents, whose life-long love, guidance, and endless sacrifices were the key factors of success throughout my life. Thanks also go to my brother and sister for their encouragement.

AN ABSTRACT OF THE DISSERTATION OF

Ayman Souheil Tajeddine for Doctor of Philosophy
Major: Electrical and Computer Engineering

Title: Reputation-Based Trust Model for Secure Routing in Wireless Sensor Networks

Sensor network technology has gained much attention in the past few years as it promises to improve data collection and statistical analysis. This technology promises to improve data collection and statistical analysis and to have an important role in pervasive computing. However, sensor nodes are severely constrained in memory, processing power, and energy resources. In addition, they are prone to several security attacks due to their wireless nature and their deployment in open and unattended areas. There are several methods to detect misbehaving nodes and provide secure routing while accounting for energy consumption and lengthening the network lifetime; among these are: reputation- and trust-based methods, location isolation, and behavior-based techniques.

In this dissertation, we present CENTERA, a CENTralized Trust-based Efficient Routing protocol with an appropriate Authentication scheme for wireless sensor networks (WSN). CENTERA utilizes the more powerful base station (BS) to gather minimal neighbor trust information from nodes and calculate the best routes after isolating different types of misbehaving nodes. Periodically accumulating these simple local observations and approximating the nodes' battery levels, the BS draws a global view of the network, calculates three quality metrics –maliciousness, cooperation, and compatibility, and evaluates Data Trust and Forwarding Trust values for each node. Based on these metrics, the BS isolates “bad” nodes, misbehaving or malicious, for a certain period of time, and put some nodes on probation. CENTERA increases the node's bad/probation level with repeated misbehavior, and decreases it otherwise. Then it uses a very efficient method to distribute the routing information to “good” nodes.

Based on its target environment, and if required, CENTERA uses an authentication scheme suitable for severely constrained nodes, ranging from the symmetric RC5 for safe environments under close administration, to pairing-based cryptography (PBC) for hostile environments with a strong attacker model. We implement CENTERA using TOSSIM and verify its correctness and show energy and data performance. CENTERA is shown to be a scalable protocol resilient to most attacks while imposing minimal overhead levels on the sensor nodes, depending on the different parameters and assumptions.

CONTENTS

ACKNOWLEDGMENTS	v
ABSTRACT.....	vi
LIST OF ILLUSTRATIONS	x
LIST OF TABLES.....	xii
Chapter	
1. INTRODUCTION.....	1
1.2. Background.....	3
1.2.1. WSN Challenges and Design Issues.....	4
1.2.2. Security Requirements.....	6
1.2.3. Background on Security	9
1.3. Dissertation Organization	11
2. LITERATURE SURVEY.....	12
2.1. Trust and Security in WSNs	12
2.2. Appropriate Authentication Techniques for WSNs.....	20
3. CENTERA - A CENTRALIZED TRUST-BASED EFFICIENT ROUTING PROTOCOL WITH AUTHENTICATION FOR WIRELESS SENSOR NETWORKS	23
3.1. Authentication Techniques	23
3.1.1. Symmetric Key Ciphers	24
3.1.2. Asymmetric Key Ciphers	27
3.1.3. Hybrid Techniques	29

3.2. CENTERA Basic Epochs	31
3.2.1. Initialization Epoch.....	32
3.2.2. Neighbor Discovery Epoch	35
3.2.3. Node Observation Epoch.....	36
3.2.4. Report Accumulation Epoch	37
3.2.5. Node Analysis and Metric Calculations Epoch.....	39
3.2.6. Bad Nodes Isolation Epoch	44
3.2.7. Basic Routing Epoch	45
3.2.8. Routing Information Dissemination Epoch.....	45
4. ATTACKS AND MISBEHAVIORS	48
4.1. External Attackers	48
4.2. Protocol Specific Attacks	48
4.3. Bad Packet Attacks	49
4.4. Packet Number Discrepancies	50
4.5. Broadcasting Nodes	51
4.6. Colluding Nodes	51
4.7. Node ID Attacks	52
4.8. False Neighborhood Attacks.....	53
4.9. Bad Mouthing and False Praising Attacks	54
4.10. Transient Behavior Attacks	54
4.11. Sinkhole Attacks.....	55
4.12. Modification Attacks	55
4.13. Packet Delay and Packet Replay Attacks	55
5. SIMULATION VERIFICATION	58
5.1. Simulation Setup.....	58
5.2. Simulation Results	59
5.2.1. Different Topologies and Sizes	59

5.2.1.1. The Linear Topology.....	59
5.2.1.2. The Tree Topology.....	60
5.2.1.3. The Grid Topology.....	61
5.2.2. Different Attacks and Misbehaviors.....	65
5.2.2.1. Uncooperative Node.....	65
5.2.2.2. Outsider Attacker and a Malicious Node	66
5.2.2.3. Counter Manipulating Nodes	68
5.2.2.4. Impersonator Node	69
5.2.2.5. Broadcasting Node	69
5.2.2.6. A Comprehensive Case	70
5.3. Authentication Benefits	72
5.3.1. Symmetric Cipher – RC5	72
5.3.2. Asymmetric Cipher – PBC.....	73
5.3.3. The Hybrid Techniques	75
6. ENERGY CALCULATIONS	76
6.1. Communication Overhead without Authentication.....	76
6.2. Communication Overhead with Authentication	81
6.3. Overhead with respect to Activity Period.....	85
6.4. Authentication Overhead.....	86
6.5. Authentication Advantage – A Broadcasting Node.....	87
6.5. Network Lifetime Calculations.....	89
7. CONCLUSION AND FUTURE DIRECTIONS.....	92
REFERENCES	95

ILLUSTRATIONS

Figure		Page
3.1.	Authentication Categories, WSN Requirements, and Decisions.....	24
3.2.	Flowchart of the Different Epochs in the BS.....	31
3.3.	Flowchart of the Different Epochs in Any Node.....	32
3.4.	The Initialization Epoch Steps.....	33
3.5.	Different Message Formats.....	34
5.1.	The Linear Topology.....	59
5.2.	The Tree Topology.....	60
5.3.	The Grid Topology (9x9).....	61
5.4.	Initial Routing Paths in the 9x9 Grid Topology (Hop Cost = 1).....	62
5.5.	Routing Paths with hop cost equals zero.....	63
5.6.	Routing Paths.....	64
5.7.	Uncooperative Node 12.....	65
5.8.	The isolation of the broadcasting node 23.....	70
5.9.	The Initial Routing Paths with the Attackers.....	71
5.10.	The Updated Routing Paths.....	72
5.11.	The Updated Routing Paths and Node Isolations.....	74
6.1	Communication Overhead without Authentication.....	79
6.2.	Overhead with respect to Normal Data.....	79
6.3.	Energy Overhead without authentication in uJ.....	80
6.4.	Communication Overhead with Authentication.....	83
6.5.	Overhead with respect to Normal Data.....	84
6.6.	Energy Overhead with PBC authentication in uJ.....	84

6.7.	Average Overhead with respect to the Activity Period.....	85
6.8.	Cumulative Number of Exchange Bytes in the Network up to each Pass.....	89
6.9.	Average Node Residual Energy in J.....	90
6.10.	Worst Case Residual Energy in J.....	91

TABLES

Table		Page
2.1.	Comparison among Most Popular WSN Routing Techniques.....	18
3.1.	Node Neighbor Activity Table at Node X.....	37
3.2.	Nodes Evaluation by the BS.....	42
4.1.	Summary of Attacks.....	56
5.1.	Load on the BS neighbors with respect to hop cost.....	65
5.2.	Different Values of node 23 at BS every period.....	67
6.1.	Bytes Transmitted and Received without Authentication.....	77
6.2.	Transmission Energy in ($\mu\mathbf{J}$) without Authentication.....	80
6.3.	Bytes Transmitted and Received with PBC Authentication.....	82
6.4.	Transmission Energy in ($\mu\mathbf{J}$) with PBC Authentication.....	82
6.5.	Total Number of Authenticated Bytes.....	86
6.6.	Cumulative Number of Bytes Up to each Pass.....	88
6.7.	Network Life-time Calculations.....	90

CHAPTER 1

INTRODUCTION

Wireless Sensor Networks (WSN) are a collection of sensor nodes spatially dispersed to sense and collect data from the environment and collaborate with each other to deliver their readings to a base station (BS) for statistical analysis or merely data collection [1]. Sensor nodes are unattended devices that are severely constrained in terms of processing power, memory size, and energy levels; and thus security and energy consumption are major concerns for any WSN implementation or application. Several security attacks can be launched on a WSN to disrupt its routing scheme, to broadcast false or harmful information, to drain the node battery and thus decrease the network lifetime, among others [2]. There are several methods to detect malicious or misbehaving nodes, and to provide secure routing, which is another critical issue in WSNs, while accounting for energy consumption and hence lengthening the network lifetime. Among these are reputation-based and trust-based methods [3], location isolation [4], and behavior-based techniques [5]. For the proper functioning of these schemes, a secure and efficient authentication scheme is required to validate nodes to each other and to the base station with minimal processing power and data transmission overhead.

Note that the term misbehavior or misbehaving node is used in this document to reflect a selfish node, or any other node that willingly or unwillingly, due to a malfunction or defect, interrupts or abuses the functionality of the network in any way possible, except for sending malicious packets. Misbehaviors include manipulating protocol-specific parameters, sending through improper neighbors, declaring erroneous

data, or even broadcasting or dropping packets.

In this dissertation, we present a CENTralized Trust-based Efficient Routing protocol with an appropriate Authentication scheme (CENTERA) for WSNs. Utilizing the centralized approach, CENTERA, based on CENTER [6], uses the more powerful and more knowledgeable BS to provide a more trusted network environment with more efficient and secure routing paths, while decreasing the load on the severely-constrained sensor nodes.

In CENTERA, the sink BS periodically gathers minimal observations from the individual nodes about the number of packets sent through neighbors and then performs several checks and calculations to have a better and more accurate view of the network. The BS approximates the battery level of every node based on its presumed activity and calculates several quality metrics for every node, namely the maliciousness, cooperation, and competence levels. Then, the BS evaluates two trust values for each node—namely Data Trust and Forwarding Trust.

Following the quality metrics calculations, the BS is able to detect several types of bad nodes: a malicious node sending false or illogical information, a non-cooperative node not reliably forwarding the packets of other nodes, an incompetent node unable to correctly deliver packets to the sink BS, or a malfunctioning/malicious node broadcasting packets. Those “bad” nodes are isolated for a period of time that depends on their history. Thus the sink BS increments the bad or probation level of every node with repeated bad behavior, while decrementing this level for repeated “good” behavior.

Finally, the BS uses an efficient method to disseminate updated routing information to all the network nodes such that each node knows its uplink nodes to forward its packets to the BS, and its next hop downlink node to forward its own

packets through it.

CENTERA provides a trust-based routing protocol while accounting for the severely-constrained sensor nodes batteries and preserving energy in the presence of misbehaving nodes by detecting and isolating them. CENTERA eliminates the power-consuming reputation inquiries and computations required by a distributed approach; nodes are required to send minimal additional information, namely their next hop and a counter (p_counter) for the downlink neighbor (DL) towards the BS and each uplink neighbor, showing the number of packets sent through and forwarded from this neighbor.

For the proper functioning of our routing protocol and the necessary validation of the nodes to each other and to the base station, CENTERA uses a secure and efficient authentication scheme suitable for the extremely limited sensor nodes in WSNs providing acceptable security levels while requiring minimal processing power and data transmission overhead. Based on the target environment, the attacker model, and the sensitivity of the data being collected, CENTERA uses the most appropriate authentication scheme—namely the symmetric key cipher RC5 in case of a safe environment under close administration; or the asymmetric key cipher identity-based encryption—elliptic curve cryptography (IBE_ECC) or PBC in the case of a hostile environment with a strong attacker model. The decision as to which technique to choose and the sizes of the keys and Message Authentication Code (MAC) and their installation in the sensor nodes occurs in the initialization phase prior to launching.

1.2. Background

In this section, we present the main challenges and design issues facing WSNs and their general security requirements. We also present a brief background on security

as some cryptographic mechanisms constitute an essential part of CENTERA in the authentication schemes used and the BS decisions.

1.2.1. WSN Challenges and Design Issues

As previously mentioned, wireless sensor nodes are extremely limited devices in memory, energy, and processing powers. So, one of the main challenges to keep in mind while designing a routing protocol for WSNs is to assure proper data communication while preserving the sensor nodes' resources and thus extending the network lifetime. In the following, we present a list of the most common routing and design challenges for WSNs [7,8,9]:

- **Node Deployment:** Depending on the application, sensor nodes may be manually deployed in a deterministic way in the network or randomly scattered. This affects the routing protocol performance as to choose the best and most efficient routes.
- **Energy Consumption without Losing Accuracy:** This may be one of the most important challenges in the severely-constrained WSN, for all protocols, computations, and even data exchanges must be extremely energy efficient while properly performing their role.
- **Data Reporting Model:** The implemented routing protocol functionality is highly affected by the data reporting method used in the application; whether query-driven, event-driven, time-driven, or a hybrid combination of them.
- **Node/Link Heterogeneity:** Depending on the applications, nodes may be homogeneous or heterogeneous in terms of capabilities and available resources.

- **Fault Tolerance:** The routing protocol must be resilient to the failure of some sensor nodes and formulate new routes to keep the network functioning properly.
- **Scalability:** The routing protocol for WSNs must be able to deal with the huge number of sensor nodes in the order of hundreds or even thousands. Also, it should be able to accommodate the sudden increase of packets as an event occurs.
- **Network Dynamics:** Most of the literature studies assume that the nodes in a WSN are stationary; however, some applications may necessitate the mobility of the sensor nodes and sometimes the BS.
- **Connectivity:** For the proper operation of the WSN, sensor nodes need to be highly connected, and this depends on the high node density and the random node distribution.
- **Coverage:** As each sensor node is limited in the physical area it can access, a key design parameter in WSN is the area coverage.
- **Data Aggregation:** Depending on the application, sensor nodes may aggregate and combine data according to a certain function such as duplicate suppression or average. Data aggregation is used to reduce redundant communicated data.
- **Quality of Service:** On top of the strict energy requirement for WSNs, some applications may force certain quality of service requirements such as delivering data within a time frame.
- **Transmission Media:** This point dictates the design of the medium access control (MAC) layer depending on the problems of the wireless medium. This point is outside the scope of this dissertation and included only for

completion.

All of these challenges are amplified in WSNs due to the strict constraints imposed by a sensor node, whether in terms of processing, energy, or memory.

In addition to all of these design issues, there remains the issue of security to be discussed in the next section.

1.2.2. Security Requirements

Given the severely constrained nature of the sensor nodes, securing data and resources tends to be a major challenge in WSNs. We will start by presenting a list of the most common attacks and misbehaviors that may target WSNs and the implemented trust protocols [2, 10-16]. Each attack is categorized as an external attack from an outsider, an inside attack from a node within the network, or could be initiated either way:

- **Flooding Attack or DoS Attack: (Internal or External).** The attacker, whether malicious or malfunctioning floods the victim nodes or the network as a whole and depletes the nodes limited resources. A special type of this attack is the Hello Flood Attack.
- **Energy Drain Attack: (Internal or External).** In an attempt to drain the energy of a node, the attacker requires it to respond to a large amount of traffic.
- **Sybil Attack: (Internal or External).** This attack is imposed by a node assuming several identities.
- **Packet Injection: (Internal or External).** The attacker injects packets with false data into the network.
- **ID Spoofing: (Internal or External).** The attacker spoofs the source ID and

sends a routing disrupting packet.

- Traffic Analysis: (External). The attacker analyzes the traffic to understand the nature and topology of the network. It also can locate the BS and most critical nodes in the network in order to attack them.
- Selfishness: (Internal). A node may be selfish and choose not to participate in the routing of the network. A common reason may be the low remaining battery power. However, even in that case, selfishness may affect the whole network.
- Colluding Nodes: (Internal). Advanced attacks may need the collusion of two or more nodes to be performed and greatly affect the network.
- Sinkhole: (Internal). The malicious attacker advertises fake routing information to draw traffic towards itself without forwarding it.
- Black-hole, Grey-hole: (Internal). Similar to the sinkhole attack, however in this case the attacker is refusing to forward all or part of the normal traffic it receives.
- Wormhole Attack: (Internal). The attacker creates an out-of-band channel to replay messages to another part of the network.
- Replay Attack: (Internal). Resending routing or other messages at a later time.
- Packet Delay: (Internal). The attacker delays forwarding a received packet to a later time.
- Modification Attack: (Internal). The attacker tampers with the packets it is forwarding (routing information, data...).
- Routing Loop: (Internal). This attack may be a special case of the previous

one, in which the attacker changes the routing information to cause a routing loop.

- On-Off Attack or Transient Behavior Attack: (Internal). In an attempt to be undetected, the attacker alternates in behaving good and bad at several rates.
- Conflicting Behavior: (Internal). Similar to on-off attack, however in this case the attacker is alternating its good and bad behaviors based on the its different peers.
- Intelligent Behavior Attack: (Internal). Similar to the previous two attacks, however this time the attacker is intelligent to choose to selectively provide good or bad services based on the trust rating threshold.
- Whitewashing: (Internal). After having a bad reputation, the attacker leaves the network and enters again with a new identity to have a fresh reputation value and have all of its wrong doings wiped off.
- Bad Mouthing and False Praising: (Internal). The attacker gives dishonest recommendations to meet its own personal interest. It provides good peers with bad reputation values while boosting up the reputation values of malicious peers. This attack can only be performed when indirect trust is used.

As for the security requirements in a WSN, Lopez et.al exhaustively summarized them in [13] as follows:

- Confidentiality: Data should be understood only by its intended receiver.
- Integrity: Data should not be altered from sender to receiver.
- Authentication: Allows the receiver to verify the true sender of the data.

- Authorization: gives privilege to perform certain operations.
- Availability: services should be accessible whenever needed.
- Freshness: confirms that data is recent.
- Forward and Backward Secrecy: newly added nodes should not be able to read older messages and current nodes should not be able to read future messages when they leave the network.
- Self Organization: being an ad-hoc kind of network, nodes must have the ability to react, organize and heal themselves autonomously.
- Auditing: sensor network elements must be able to store any event occurring within the network.
- Non-repudiation: confirms that a node has sent a message without the ability for it to deny.
- Privacy and Anonymity: hiding and protecting the ID and the location of nodes.

Note that this list is exhaustive and application dependent showing all the requirements an application may require. Some requirements are very strict and resource consuming to be applied in the WSN constrained environment unless specifically required.

1.2.3. Background on Security

In order to secure any computer network and to properly perform the network services, several basic cryptographic mechanisms must be put into action. In this subsection, we present a brief overview of the main cryptographic mechanisms [17] used in our work; mainly encryption, hashing, and message authentication code (MAC).

In order to provide confidentiality and prevent data from being leaked, an

encryption algorithm with the aid of a secret key is used to generate a cipher text or encrypted data in unintelligible form. The receiver also uses a key to decrypt to reproduce the plain text and understand the initial message. Encryption is categorized into two main approaches: symmetric-key encryption and asymmetric key (or public key) encryption. In the former both the sender and receiver share the same key to perform the encryption and decryption of the data. This key must be kept secret by the sender and receiver.

In contrast, asymmetric key encryption utilizes a key pair: a public key and a private key. Any key can be used to encrypt data and the other will be used to decrypt it. In public key cryptography, one key is kept private with its owner and the other is public (available for all). Depending on the way the keys are used, public key cryptography can provide confidentiality or authentication. If the sender used its private key to encrypt the message, any receiver can decrypt the message using the published sender's public key; however as no one has access to the private key, the receiver can authenticate the message as being sent by its sender. On the other hand, if the sender used the public key of the recipient to encrypt a message, only the intended recipient can decrypt it and understand its content. This is confidentiality. Asymmetric key cryptography requires larger keys and complex mathematical computations making it harder to use in constrained environments.

A major category in cryptographic mechanisms is hashing and MAC functions used to provide data integrity and authenticity. In general, these mechanisms use a transformation function to produce a unique output string (called the message digest) of fixed length to any input message. This digest is appended to the end of the message to prove that the received data is correct and was not tampered or changed through the communication channel. The main difference between a MAC and a hash function is

that the former uses a secret key to generate its message digest, thus providing an extra sense of authentication.

1.3. Dissertation Organization

The rest of the dissertation is organized as follows: Chapter 2 surveys the previous work in the area of trust-based routing protocols and authentication techniques used in wireless sensor networks. Chapter 3 presents the system with its different epochs, definitions, and parameters. Chapter 4 explains a list of attacks and misbehaviors and analyzes the methods of detection by CENTERA. TOSSIM simulation verifications are presented in Chapter 5. Chapter 6 presents the energy calculations of CENTERA. Finally, Chapter 7 presents some conclusions and future work.

CHAPTER 2

LITERATURE SURVEY

Sensor network technology promises to improve data collection and statistical analysis and to have an important role in pervasive computing. However, sensor nodes are severely constrained in memory, processing power, and energy resources. In addition, they are prone to several security attacks due to their wireless nature and their deployment in open and unattended areas.

2.1. Trust and Security in WSNs

In this section, we will discuss the literature related to WSN trust and security. The authors of [1] explain the difference between sensor networks and traditional wireless *ad hoc* networks. For each layer of the protocol stack, they survey the different issues and technologies for the sensor networks and provide the available solutions for each. They also highlight the open research issues at each layer.

The authors of [7] discuss the challenges in designing a routing protocol for wireless sensor networks and provide a survey of the different available routing techniques. They classify the techniques based on the network structure as flat, hierarchical, or location-based; and based on the protocol operation as query-based, multipath-based, quality-of-service-based, coherent-based, and negotiation-based. For each routing technique, they state its advantages and shortcomings and study the energy and communication overhead tradeoffs.

The authors of [2] consider routing security. They discuss the different attacks and present countermeasures. They analyze the security of the major routing protocols

and energy-conserving topology maintenance algorithms for sensor networks. They explain how attacks can be adapted from peer-to-peer and *ad hoc* networks to sensor networks, and present two new attacks: sink holes and HELLO floods.

An energy-efficient routing technique is discussed in [18], where the authors develop and evaluate many techniques to enhance routing based either on energy histograms or solely on localization. They show the network lifetime gains for each technique. Localization is a method that can be beneficial to detect complex colluding nodes attacks.

Several methods are available to detect misbehaving nodes, to lengthen the network lifetime, and to decrease energy consumption while providing secure routing; among these are: reputation- and trust-based methods [3,19], location isolation [4], and behavior-based techniques [5]. Of these, we focus mainly on trust- and reputation-based methods, with some behavior consideration at the BS.

The authors in [19] explain the difference between sensor networks and traditional mobile *ad hoc* networks and survey several trust-based systems in wireless sensor networks. They provide different trust definitions and properties as defined in the literature and state that the definition is application-specific and depends on the methods used to calculate the trust value. They extend the definition of trust to include the sensor data and reliability as a new component and introduce a new trust model that is believed, as per the authors, “to be very robust as it addresses all the drawbacks from the existing approaches.”

The authors in [20] differentiate between the security requirements of WSNs and other networks. They present iTrust, which depends on the presence of monitor nodes to assess the behavior of their neighbors and distribute the trust indices after each session. The authors evaluate their model and show its robustness against different

attack scenarios. This method uses the available constrained sensor nodes to assess behavior and distribute trust, instead of fully delegating this burden on the powerful BS.

Trust models are surveyed in [21]. The author explains the difference between security and trust and between reputation and trust. He surveys the methodologies and factors used in the different trust models and states that in wireless sensor networks it is not enough to examine routing messages to infer trust; however, new methods are needed to calculate both communication trust and data trust while keeping in mind that data is sometimes continuous.

In [22] the authors present a security survey on WSNs. They first target the network layer and identify the vulnerabilities and summarize the different defense methods in this protocol layer. Then, they divide the general security issues into seven categories, namely: cryptography, key management, attack detections and preventions, secure routing, secure location security, secure data fusion, and others. In this division, they summarize the different techniques used and point out their pros and cons.

The authors in [23] assure that trust is an important factor for WSNs security schemes and provide a detailed study on trust mechanisms and attacks and countermeasures. They provide a categorization of all trust-related attacks in WSNs. They also analyze the different trust schemes and illustrate the differences and challenges of each. In their work, they provide an extensive literature survey of trust mechanisms used in WSNs and they present open research directions.

In [24] the authors focus on the energy limitations of the sensor nodes and propose a centralized energy-efficient routing protocol for WSNs to reduce energy consumption and thus, increase network lifetime. Their protocol, called Base-Station Controlled Dynamic Clustering Protocol (BCDCP), increases network lifetime and average energy savings by evenly spreading the energy dissipation among all nodes.

Although this work does not target trust or security, it is of interest since they highlight the energy saving benefits of a centralized routing scheme, on which we focus our work.

The authors of [54] propose TrustMIX incorporating computational trust management into a previously proposed WSN scheme for data gathering known as MIX. Their main motivation and result are to target and reduce the effect of sinkhole attacks that MIX is vulnerable to, without affecting the increased network lifetime offered by it. TrustMIX is a simple and fully distributed algorithm that secures data by letting them avoid massive sinkhole attacks. The authors study the effect of incorporating trust into MIX on energy consumption. They show that for a small number of sinkhole attackers, low energy consumption is preserved for all nodes; however for a larger set of simultaneous attackers, only close nodes to the BS have low energy consumption while peripheral nodes away from the sink deplete their energy faster. The authors also stressed the importance of time in detecting an attacker as more attackers need more time to be detected by the trust engine.

In [55], the authors stress the importance of security in WSNs while maximizing overall data reliability and minimizing energy consumption. They address these issues by applying their Security Adaptation Reference Monitor (SARM), which supports both context monitor and behavior control to offer a tradeoff between energy consumption and security. They implement their security scheme in the kernel to offer global security for any application instead of the layered security mechanisms. They simulated SARM using AnyLogic under sinkhole attacks and show that it is reliable, secure, and power efficient, even at 50% of attackers. Thus the WSN version of SARM constitutes a good platform that detects any sinkhole within the Base Station and removes its connections.

The authors in [56] propose a trust aware routing framework (TARF) to secure WSN routing against routing information replay attacks. TARF decreases the effect of such attackers while keeping an acceptable energy overhead. In TARF, the node's trustworthiness is included in the routing decisions to allow nodes to avoid attackers using forged identities acquired through replaying. The authors show empirical and simulation results to prove TARF's satisfactory performance in routing while detecting routing information replay attacks.

In [57], the authors address the issues of reliability and security in WSNs by proposing a weighted energy-aware routing protocol (WEAR) as a general energy-efficient, fault tolerant, load balanced, and scalable routing protocol. They argue that none of the literature work simultaneously satisfies all of these four requirements. WEAR incorporates four factors namely the distance to the destination, the energy level of the sensor, the global location information and the local hole information. The authors define a hole to be "a large space without active sensors caused by dead or faulty sensors" and propose another protocol to handle holes, identifies them, and maintains their information in a dynamic WSN. Finally they provide a comprehensive simulation and show the superiority of WEAR over GEAR and GPSR in terms of eight performance metrics they propose.

Several surveys discussed comprehensively the different routing techniques in WSNs [2,7,25–31]. Table 2.1 presents a comparison between the most popular routing techniques in WSNs based on common attributes.

The different parameters included in Table 2.1 are explained in the following list:

- Classification: Routing protocols are classified based on the network structure into flat-based routing containing nodes with equal roles;

hierarchical-based routing containing nodes with different functions; and location-based routing where routing depends on nodes' positions.

- Mobility: Sensor nodes may be considered as stationary or mobile with the possibility to allow for limited mobility.
- Position Awareness: A sensor node is aware of its spatial position.
- Power Usage: Indicates the power consumption of the whole protocol.
- Negotiation-based: Such protocols initiate negotiation messages prior to data messages in an attempt to eliminate the transmission of duplicate or redundant information.
- Data Aggregation: The option to combine data from different nodes based on some function.
- Localization: The ability to estimate the location of each node.
- Complexity: Given the limited-nature of sensor nodes, applied protocols must be simple.
- Scalability: The protocol must cope with the huge number of typically available sensor nodes.
- Multipath: In an attempt to improve network reliability, some routing protocols utilize multiple paths to route packets at the expense of additional overhead.
- Query-based: In such routing protocols, the destination node queries the network for its required data, and the node with such data replies.
- Centralized/Distributed: In centralized routing protocols, a central node is responsible for the network routing, whereas in distributed protocols, each node has its own routing decision.

Table 2.1 – Comparison among Most Popular WSN Routing Techniques

	Classification	Mobility	Position Awareness	Power Usage	Negotiation-based	Data Aggregation	Localization	Complexity	Scalability	Multipath	Query-based	Centralized / Distributed
CENTERA	Flat	No	No	Very Limited	No	No	No	Low	Good	Possible	No	C
SPIN	Flat	Possible	No	Limited	Yes	Yes	No	Low	Limited	Yes	Yes	D
Directed Diffusion	Flat	Limited	No	Limited	Yes	Yes	Yes	Low	Limited	Yes	Yes	D
Rumor Routing	Flat	Very Limited	No	N/A	No	Yes	No	Low	Good	No	Yes	D
GBR	Flat	Limited	No	N/A	No	Yes	No	Low	Limited	No	Yes	D
MCFA	Flat	No	No	N/A	No	No	No	Low	Good	No	No	D
CADR	Flat	No	No	Limited	No	Yes	No	Low	Limited	No	No	D
COUGAR	Flat	No	No	Limited	No	Yes	No	Low	Limited	No	Yes	D
ACQUIRE	Flat	Limited	No	N/A	No	Yes	No	Low	Limited	No	Yes	D
EAR	Flat	Limited	No	N/A	No	No	No	Low	Limited	No	Yes	D
LEACH	Hierarchical	Fixed BS	No	Maximum	No	Yes	Yes	High	Good	No	No	D
TEEN & APTEEN	Hierarchical	Fixed BS	No	Maximum	No	Yes	Yes	High	Good	No	No	D
PEGASIS	Hierarchical	Fixed BS	No	Maximum	No	No	Yes	Low	Good	No	No	D
MECN & SMECN	Hierarchical	No	No	Maximum	No	No	No	Low	Low	No	No	D
SOP	Hierarchical	No	No	N/A	No	No	No	Low	Low	No	No	D
HPAR	Hierarchical	No	No	N/A	No	No	No	Low	Good	No	No	D
VGA	Hierarchical	No	No	N/A	Yes	Yes	Yes	High	Good	Yes	No	D
Sensor Aggregate	Hierarchical	Limited	No	N/A	No	Yes	No	Low	Good	No	Possible	D
TTDD	Hierarchical	Yes	Yes	Limited	No	No	No	Moderate	Low	Possible	Possible	D
GAF	Location	Limited	No	Limited	No	No	No	Low	Good	No	No	D
GEAR	Location	Limited	No	Limited	No	No	No	Low	Limited	No	No	D
SPAN	Location	Limited	No	N/A	Yes	No	No	Low	Limited	No	No	D
MFR	Location	No	No	N/A	No	No	No	Low	Limited	No	No	D

One thing that can be directly noticed is that out of the whole list shown in Table 2.1, only CENTERA utilizes the centralized approach. All of the surveyed previous work focuses on distributed methods to calculate trust and reputation and to provide security for the wireless sensor network. To the best of our knowledge and till the time of this writing, CENTERA is the first centralized routing protocol in WSN that

incorporates security and trust criteria in the core of the routing decision engine.

It is more logical to utilize the centralized approach in a WSN and make use of the more powerful and knowledgeable BS to perform these calculations and eliminate the burden of the power-consuming reputation inquiries and computations imposed by the distributed approach on the sensor nodes.

In a typical WSN scenario, the sensor nodes sense and collect physical data and deliver them to the BS to be used for statistical analysis. A BS is assumed to be all powerful in terms of energy, processing power, and storage capabilities. Also, it is assumed to be secure and under the direct supervision of the network administrator since if the BS is exposed then the whole network loses its purpose.

Thus, in order to decrease the load on the severely-limited sensor nodes and extend the network lifetime, our direction is to make use of the normally sent data packets after adding minimal overhead. This way we utilize the all powerful, knowledgeable, and secure BS to efficiently disseminate correct, efficient, and secure routing information, after intelligently detecting and isolating all misbehaving nodes from the network.

This direction to use the centralized approach is in harmony with the recommendation of the Open Networking Foundation (ONF), Software Defined Networking (SDN) to use centralization of network intelligence as one of the new norms for networks [32].

ONF is a non-profit user-driven organization that promotes SDN, a new networking approach and architecture that offers to separate the network intelligence and control from the task of data forwarding and move it into centralized controllers maintaining a detailed global view of the network.

2.2. Appropriate Authentication Techniques for WSNs

In this section based on [33], we review the literature related to authentication techniques used in the severely constrained wireless sensor network environments.

Several researchers prefer the use of symmetric key cryptography in the WSN limited environments. The authors in [34] conduct a very useful comparison between different cryptographic and encryption techniques using a message authentication code in WSNs. They stress the importance and increasing popularity of WSNs and the importance of the choice of a feasible MAC to use. They compare symmetric and asymmetric cryptography, different encryption techniques and hashing techniques based on different criteria, mainly processing time, energy consumption, and memory requirements. First, they argue that symmetric key cryptography is more appropriate to be used due to the limited nature of WSNs. Then, they compare the different symmetric key techniques namely hashing techniques, block cipher, and stream cipher based on security in defending attacks while keeping in mind the associated overhead. They conclude that even though hash functions offer good security, block cipher is best to be used for generating a MAC in WSNs, and they specify RC5 to be the most feasible, providing good security while consuming little energy and resources.

In [35], the authors examine the energy consumption of the different symmetric key algorithms namely the block and stream ciphers, when applied to WSNs. They used the number of CPU cycles as a measure of the computational energy cost of an encryption algorithm when they compare the different ciphers. They conclude that the block cipher, byte-oriented substitution-permutation network (BPSN), is the most recommended cipher to achieve acceptable security and energy efficiency for WSNs.

In his thesis [36], Soderlund targets energy-efficient authentication in WSNs and checks the effect of the MAC lengths on the lifetime of a sensor node. The results

of his thesis show that it is recommended to use a 32-bit MAC on both the network and the data link layers to provide sufficient security without consuming too much energy. His conclusion states that symmetric cryptography is preferable over asymmetric cryptography because it is faster and consumes less energy, while specifying Localized Encryption and Authentication Protocol (LEAP) as the most recommended symmetric key solution to be used. Note that in his thesis, Soderlund states that ECC is the most promising public key cipher that can be made more efficient by hardware support; however he discarded the use of public key cryptography in general in WSNs due to its large time overhead on every message.

Other researchers prefer the use of public key cryptography and justify the added overhead by different needs. The authors in [37] argue that public key cryptography can be more flexible as a security component, for authentication and key exchange, than the extensively investigated symmetric key cryptography. They show the supremacy of ECC over RSA in public key ciphers due to its energy savings and its use of smaller keys.

In [38], the authors state that symmetric ciphers are not scalable with large numbers of nodes, and thus, with the recent curves-based cryptographic algorithms, public key cryptography is justified to be feasible for WSNs. They discuss some attacks and countermeasures, compare RSA and ECC, and they show that ECC excels over RSA by decreasing stored and transmitted data and also the time of computations.

The authors of [39] stress the importance of authentication in WSNs and its effect on avoiding attacks and preserving the limited resources of sensor nodes. The authors address the difficulty of applying public key cryptography and propose a framework based on identity based cryptography and online/offline signatures for multicast and broadcast to authenticate multicast and broadcast messages. Their model

accounts for the communications among sensor nodes as well as outside users accessing the WSN.

In [40], the authors target secure and trusted user communication, whether node-to-node or node-to-base station communications. They present a trust protocol utilizing Trusted Computing Group and an identity-based cryptosystem. They propose their model, verify it, and present some analysis of memory requirements.

The authors of [41] propose a decentralized energy-aware key management scheme for WSNs. They use identity-based encryption that uses elliptic curve cryptography, as the most promising in energy efficiency. They evaluate and analyze the system and show that their scheme reduces the overall system energy while providing confidentiality and increasing availability even when multiple nodes and stations are compromised.

It is clear that there are two main cryptographic techniques that are the symmetric-key (shared secret key) cryptography and the asymmetric key (public-key) cryptography to choose from when targeting authentication in any routing protocol or trust model in WSNs.

CHAPTER 3

CENTERA - A CENTRALIZED TRUST-BASED EFFICIENT ROUTING PROTOCOL WITH AUTHENTICATION FOR WIRELESS SENSOR NETWORKS

Our proposed routing protocol CENTERA implements a centralized trust-based routing protocol with an appropriate authentication scheme for WSNs placing most of the computational load on the more powerful sink BS. Constructing the global view of the network from minimal local information of the authenticated sensor nodes, the BS is responsible for calculating the nodes' trust information and distributing the routing information after isolating the "bad" nodes. CENTERA is divided into eight functional epochs ensuring the creation of a secure, trusted, and efficient wireless sensor network environment.

This chapter is divided into two main sections: section 3.1 discusses the authentication techniques used in CENTERA, and section 3.2 explains the eight different epochs of CENTERA in details.

3.1. Authentication Techniques

In this section, we address the three main fields in authentication that are based on symmetric cryptography, asymmetric cryptography, and hybrid techniques using both cryptographic methods. We choose the most appropriate cipher or technique from each category and discuss it briefly and highlight its advantages and disadvantages and describe the parameters affecting the WSNs environment.

A summary of the analysis presented in this section, including the main authentication categories, the WSN requirements, and the choice dependencies is

illustrated in Figure 3.1.

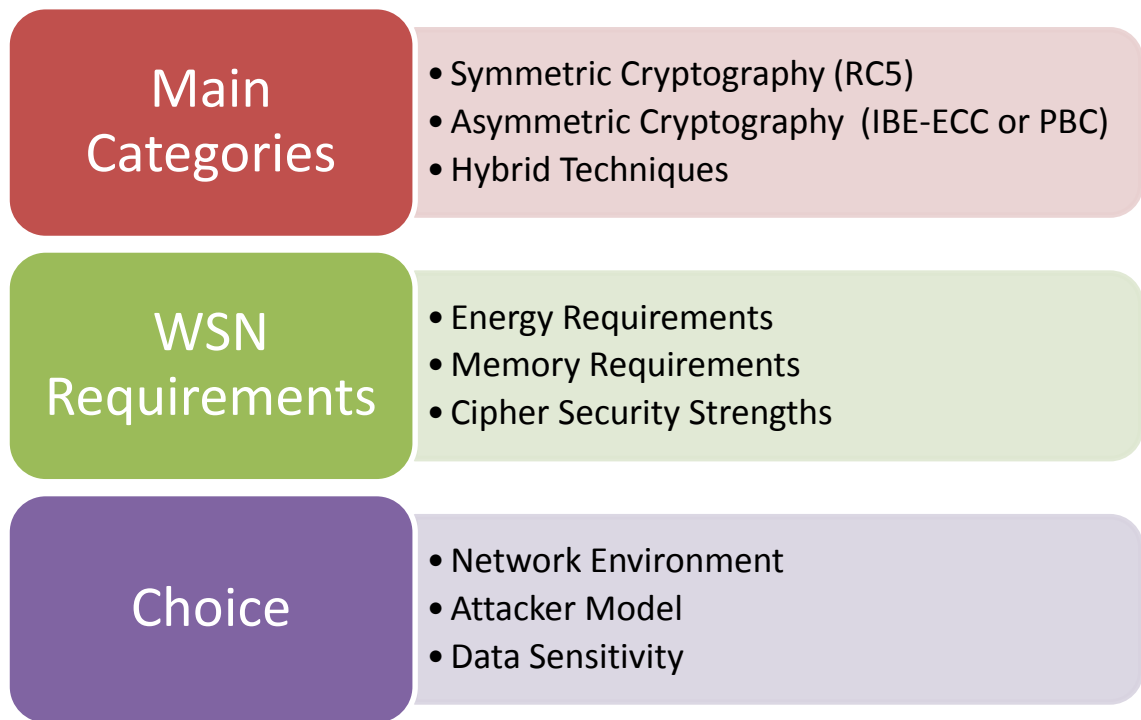


Fig. 3.1 – Authentication Categories, WSN Requirements, and Decisions

3.1.1. *Symmetric Key Ciphers*

Symmetric key encryption is the type of cryptography that uses a single key, called a secret, to encrypt/decrypt a message by the sender/receiver. In that sense, the sender and the receiver are equal entities sharing a common secret key to exchange encrypted messages and be able to decrypt them.

When compared to other types of cryptographic techniques, mainly public key, symmetric key techniques use less computation, processing, and energy; so it is more widely used for generating a MAC and ensuring authentication in WSNs. Algorithms that use symmetric keys are typically orders of magnitude faster than public key cryptography algorithms [42].

Symmetric key cryptography is divided into stream or block ciphers depending on the way the plaintext is processed; stream ciphers encrypt messages one byte at a time; whereas block ciphers encrypt blocks, which are a fixed number of message bytes considered together as a single unit.

Stream ciphers are more prone to attacks than block ciphers; and thus block ciphers are more secure and hence have a broader range of applications. Some examples of block ciphers include DES, AES, RC5, Skipjack, Puffin, and BSPN. According to [34,35], RC5 and BSPN are the best block ciphers recommended to be used in WSN environments.

We will focus our discussion on RC5 since it is more widely used and tested. RC5 is a symmetric block cipher that is suitable for both software and hardware implementations. It is very simple and fast and requires low memory, while offering good security [43].

The RC5 algorithm is a Feistel-like network making use of data-dependent rotations and modular additions and XORs. RC5 uses variable block sizes (default = 64 bits), key sizes (default = 128 bits), and number of rounds (default = 12) [43].

The benefits of RC5 can be summarized as follows:

- RC5 is much faster and suitable for WSN scenarios than other symmetric algorithms such as DES or AES and even RC4 and IDEA [34]. Being faster means it needs less processing time and thus consumes less energy.
- RC5 uses less memory than other techniques and even hashing techniques, which need higher overhead [42].
- Although RC5 consumes more energy and requires more memory than some algorithms such as Skipjack and XXETA, this difference is due to the key size and round numbers and is minor compared to the added security it

provides [44].

Thus RC5 is the most feasible to be used for WSN scenarios since it strikes the best balance as it consumes less energy than most other algorithms while providing better security than algorithms with less energy consumption.

Using RC5, there are two parameters that affect the limited nature of sensors and thus the lifetime of the WSN: 1) the key size that affects memory and computational overhead requirements, and 2) the MAC size since the MAC will be added to a message, thus increasing the message byte count and as a result, affecting the transmission and reception times and hence the energy requirement.

Accordingly, depending on the sensitivity and nature of the application of the WSN, and the hostility of the environment in which the network is deployed, and prior to launching, the administrator of the network should consider these factors and decide upon the key size and MAC length. By increasing the size of these two parameters, the RC5 overhead increases, but the algorithm gets more secure and harder to break.

The main drawback of the use of symmetric key is that both the sender and receiver share the same secret key to perform encryption and decryption. This drawback gets worse in the case where all the nodes in the network share the same key to send their readings periodically to the base station. This approach provides some authentication that the sender is a member of the network, assuming the case of a weak attacker that cannot completely take over a node. Also using this scheme, there cannot be accountability of the exact node that sent false or malicious data into the network. This highly affects trust and reputation schemes and routing protocols, since one malfunctioning node can jeopardize the whole network with no way to point out, punish, or ban such a node.

Another drawback of symmetric key cryptography is that such schemes do not

scale well with large numbers of sensor nodes [38]. Thus, in some scenarios, symmetric key cryptography may just not be enough to guarantee the normal and correct functioning of the wireless sensor network; and so the need for other types of cryptography is required.

3.1.2. Asymmetric Key Ciphers

Although RC5 and symmetric key cryptography is the most logical and efficient scheme to be used in such limited-resource sensor networks, in some cases, additional security is needed even at the cost of sacrificing additional energy and memory.

As discussed earlier, in critical networks sensing very sensitive information gathered from hostile environments, it is required to have additional security imposed on the network and to be able to point out a malicious or malfunctioning node and isolate it from the network.

Also, for trust-based routing protocols, it is fundamental and critical to authenticate the source node of the message being sent in order to be able to hold each node responsible for its malicious acts or malfunctions. Thus for the well being of the network and the sensed data, it becomes logical to sacrifice some energy and memory and thus to use asymmetric key cryptography.

Asymmetric key cryptography or public key cryptography involves the use of two keys:

- A private key that is secret to the entity that needs to sign messages and to decrypt messages transmitted solely to it.
- A public key, which is known by every other node that needs to communicate with the entity, either to encrypt a message directed to the

entity or to decrypt and verify the entity's signatures.

In that sense, the term “asymmetric” shows that a key that encrypts a message or verifies a signature cannot and is not able to decrypt a message or create a signature. The two keys in public key cryptography are mathematically linked; however it is computationally infeasible to find one key knowing the other and the algorithm. Also, any of the two keys can be used for encryption and the second for decryption.

Asymmetric key cryptography produces more memory overhead, and consumes more processing power than symmetric key algorithms due to the large key sizes. With severely constrained devices such as sensor nodes, the use of public key cryptography was thought to be impractical and not suitable at all until Gaubatz et al. [45] challenged this assertion by proposing a hardware-assisted approach for such cryptography.

Then, came other public key cryptographic algorithms based on curves, mainly the elliptic curve cryptography (ECC), first introduced by Koblitz [46], that offers the same security with a smaller key size. Thus ECC consumes less memory and processing energy while maintaining the same security level, properties that are well suited for WSNs. ECC with key sizes of 160 bits has the same efficiency as that obtained by RSA with 1024 bit keys [38], and consumes five times less energy [47] while achieving the same security.

A challenge that faces asymmetric key cryptography, including ECC, is to distribute the public keys of each sensor node to other nodes. To address this problem, identity-based encryption (IBE) comes into play by simply making the public key of every entity the same as its name [48]. An extension of IBE is pairing-based cryptography (PBC), which is a practical implementation of ECC and seems to be the most suitable solution for the limited sensor nodes in WSNs [41].

Using IBE-ECC or PBC as a solution for scenarios requiring asymmetric key authentication, there are two parameters that affect the limited nature of sensors and thus the lifetime of the WSN: 1) the key size affecting memory and processing overhead requirements, and 2) the MAC size acting as a message overhead that increases the transmit and receive times and hence the energy requirement.

Using similar reasoning for the symmetric key ciphers case, depending on the sensitivity and nature of the application of the sensor network and the hostility of the environment in which the network is deployed, the administrator decides upon the key size and the MAC length. By increasing these two parameters, the algorithm becomes more secure and harder to break.

3.1.3. Hybrid Techniques

To gain the benefits of both worlds, a hybrid technique can be used in which a symmetric key cipher, such as RC5, is used for the data communication between nodes, and asymmetric key ciphers, such as PBC or IBE-ECC, is used to refresh the master key and generating a new symmetric key to be used by the sensors in the network.

In this method, there is a different set of parameters that need to be taken into account before choosing the technique and designing the system. These parameters directly affect the lifetime and functionality of the WSN:

1. Additional overhead of sending periodically a new symmetric master key by the base station.
2. Additional memory overhead: Two keys (the symmetric and asymmetric) are stored, instead of only one key as in either of the two cases above.
3. The refreshing interval: as this parameter increases, the overhead improves (decreases) while the security and capability to quickly detect attackers gets

worse (decreases).

4. The processing power by the cipher computation is in between the two previous cases, i.e. less than the asymmetric key and more than the symmetric key ciphers.

Hybrid authentication seems to be a promising and interesting technique to be used in WSNs. A more comprehensive analysis to study the different parameters and their effects on the general functionality, security, and overhead of the system is left as future work.

3.2. CENTERA Basic Epochs

In this section, the eight basic epochs of CENTERA are discussed in detail. Figures 3.2 and 3.3 illustrate the flowcharts of the different epochs with the different actions and decisions that take place within the BS and the sensor nodes, respectively. It is clear that the BS follows mostly a different flowchart with different epochs, decisions, and actions than a normal sensor node. This is due to all of the duties, intelligence, and decisions taken by the BS.

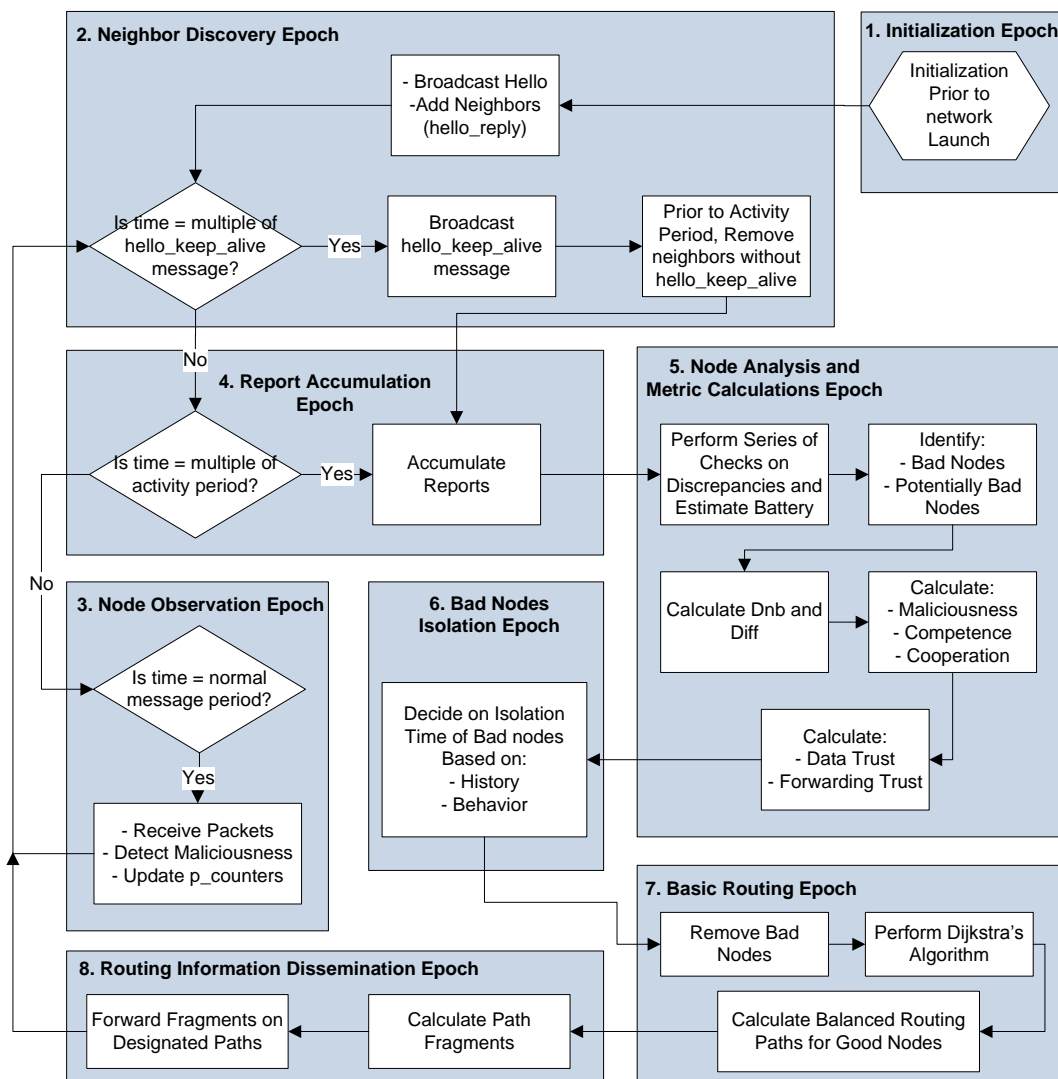


Fig. 3.2 – Flowchart of the Different Epochs in the BS

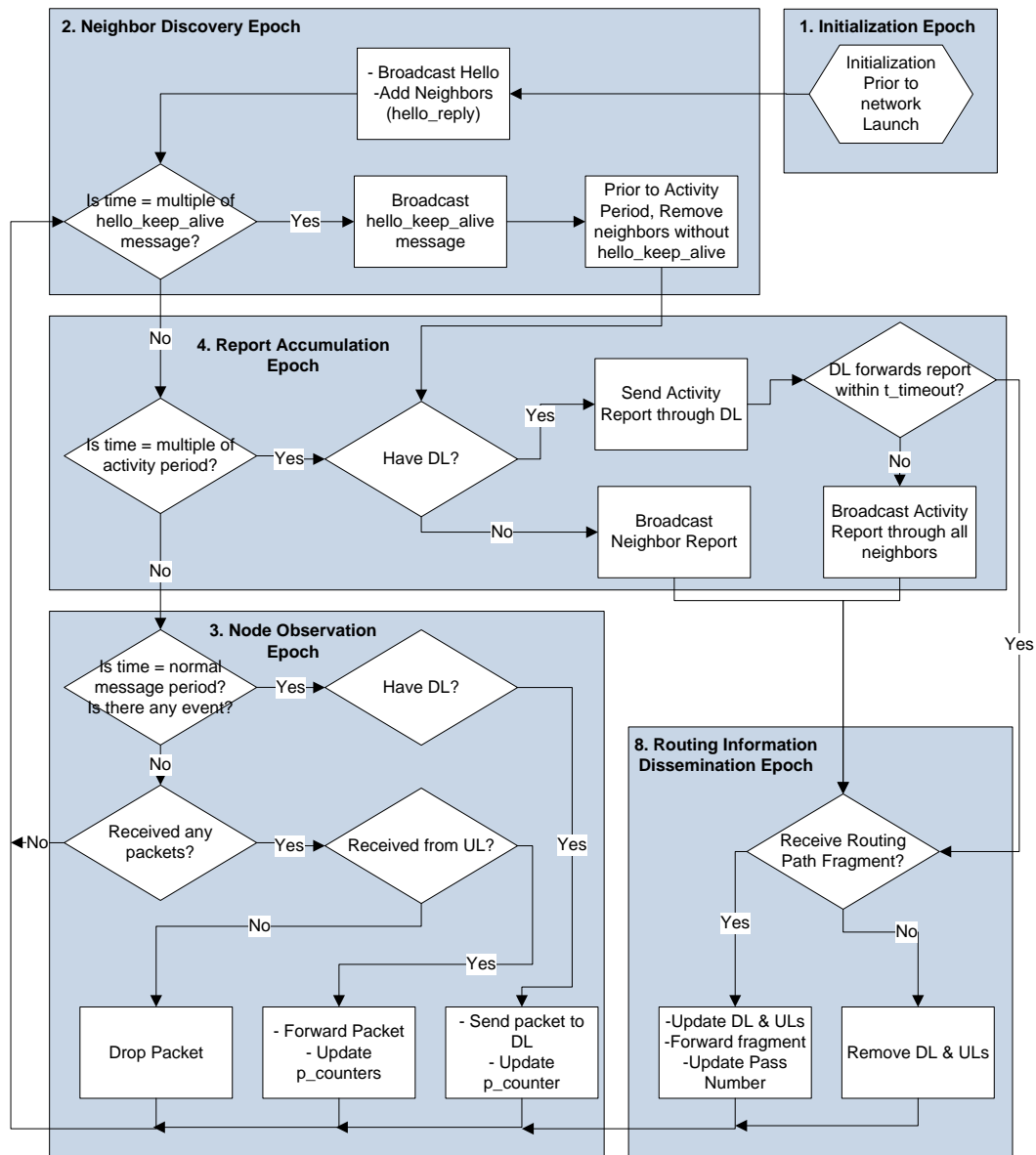


Fig. 3.3 – Flowchart of the Different Epochs in Any Node

3.2.1. Initialization Epoch

Initially and prior to network launching, the WSN administrator must study the network specifications and needs, including the expected size of the network, the nature of the application, the sensitivity of the data being sensed, and the hostility of the environment where the network will be deployed, among others. Based on the results, the administrator decides on the different parameters for the functioning of the system.

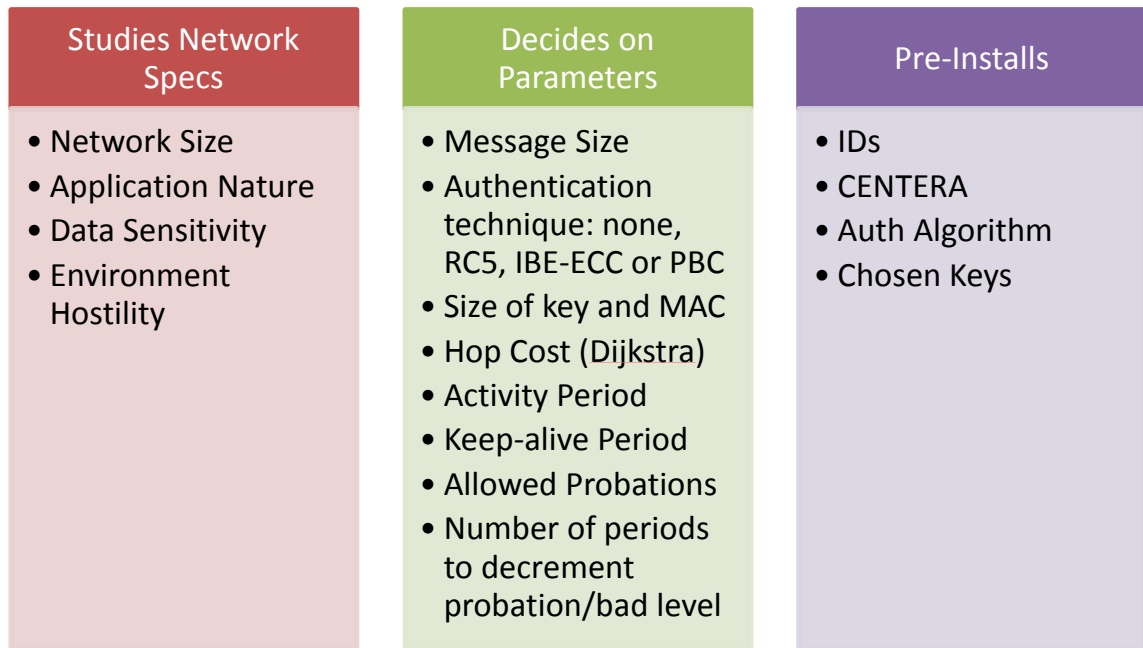


Fig. 3.4 – The Initialization Epoch Steps

Among the decisions is the message size that depends on the number of nodes in the network; i.e., any network having less than 256 nodes requires an ID size of 8 bits, whereas the ID size must be greater than that for a network with more than 256 nodes.

Another very important decision is the authentication technique, if any, to be used in the network. Depending on the hostility of the network and its unattended environment, the administrator chooses to use a strong asymmetric authentication system, like IBE-ECC or PBC, a lighter symmetric system such as the RC5, or not to use any authentication at all. The administrator decides on the sizes of the used keys and appended MAC, considered x bytes in size. Following this decision, the administrator creates and installs the network master key in all the nodes, in case of the RC5 symmetric cipher choice, or each node's unique private key, in case of the PBC choice.

Other parameters include the hop costs used in Dijkstra's algorithm that runs

on the BS, the activity period to send an activity/neighbor report from node to BS, the keep-alive period, the number of allowed probations before a node is considered bad, and the number of periods to decrement the number of probations or level of bad behavior. These parameters will be explained in more detail in subsequent sections. Of course, in this epoch the administrator fixes all the identities, the chosen authentication algorithm, and all the required algorithms and parameters for the proper functioning of CENTERA.

A summary of the different steps performed in this epoch is presented and classified in Figure 3.4.

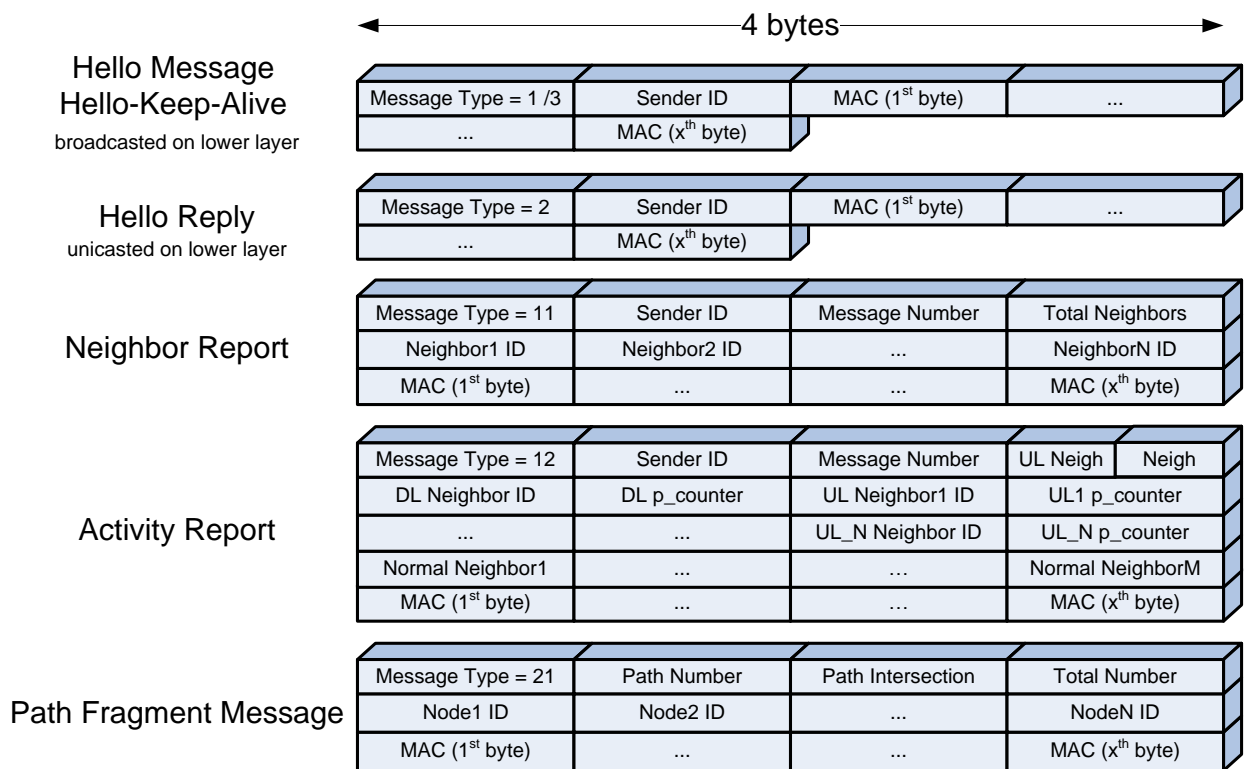


Fig. 3.5 – Different Message Formats

3.2.2. Neighbor Discovery Epoch

On network bootstrapping or with the introduction of every new node, the Neighbor Discovery Epoch is activated. In this phase, every node signs and broadcasts a one-hop hello message to introduce itself to its neighboring nodes. The hello message has the following format—{Message Type = 1 [one byte], Sender ID [one byte], MAC [x bytes]} as shown in Figure 3.5.

Note that the value of x is chosen in the Initialization Epoch by the administrator based on the network security requirements. Upon receipt of the hello message, each node within radio range checks the authenticity of the packet, if applicable, by verifying its MAC using the sender's public key (node ID) as the verification key for PBC, or using the symmetric key for RC5, as set by the administrator in the Initialization Epoch. If the packet is authentic, the receiver node adds the sender to its neighbors list and replies using a signed "unicast" hello-reply packet back to the sender in order to confirm the neighborhood between them. The hello reply message has the same format as the hello message with the Message Type = 2.

Also, every fixed time interval, set and synchronized by the BS, all nodes broadcast to their neighbors a hello_keep_alive message. The period is set by the administrator in the Initialization Epoch, and it is chosen to be a multiple of the period of sending the activity reports. This hello-keep-alive message has the exact same format as a hello message with the Message Type = 3, and is used to make sure that a node still exists in order to keep the BS updated with correct information. Upon receipt of a hello-keep-alive message, the receiving node just refreshes the status of its neighbors, without replying. Note that in case an authentication scheme is used, all types of hello messages must be signed and verified for trusted neighbors' identities.

3.2.3. Node Observation Epoch

This epoch is executed when nodes are sending normal messages of sensed data to the BS. As a requirement for this epoch, each node X keeps a neighbor activity table to store the number of communicated packets to or from each of its neighbors in a certain period, the activity period discussed later. This table contains each neighbor ID, a counter value (*p_counter*), a flag identifying the uplink (UL) nodes and another flag identifying the downlink (DL) node. In case of an UL neighbor, the *p_counter* keeps track of the number of packets that node X forwarded from this neighbor through its DL node. In case of the DL neighbor, the *p_counter* indicates the total number of packets that node X sent and forwarded; in fact the actual number of packets initiated by node X is the difference between the *p_counter* to the DL neighbor and the sum of the *p_counters* of the UL neighbors.

Note that the two flags are extracted from the path fragment message sent by the BS (explained shortly). Also note that a node sends its packets to the BS only through its DL and drops any packet received from a node other than its UL nodes.

However, before the Node Observation Epoch can be initiated and every node can start forwarding its data to the BS, the Activity Report Accumulation Epoch should be initiated for the node to know its downlink neighbor.

Table 3.1 shows an example of a Node Neighbor Activity Table, where node X has node Y as its DL and nodes Z and U as its ULs; node V is just a neighboring node without any interactions with node X. Node X forwarded 4 packets for node Z and six packets for node U. In total node X sent 15 packets through its DL node Y, and thus it initiated five packets.

Note that in case of an authentication scheme set, every normal message communicated in this block must be signed by its initiator to be verified in every hop

until the BS is reached. Also, every node forwarding this message encrypts the signature by its own key for its neighbor to verify that it is receiving a packet from an UL neighbor. Thus each node on the path decrypts the signature using the ID of its UL neighbor, and verifies the signature from the source then encrypts the source's signature by its own ID and forwards the packet. Any wrong signature causes the packet to be dropped.

Upon receipt, the BS checks the authenticity of the packet in case authentication is applied, and then increments the number of packets received from the packet source. The BS then analyzes the packet and if it is found to be malicious, it increments the number of bad packets received from this source.

Note that a bad packet is any packet containing malicious data or code intended to cause any kind of harm in the sensor nodes or the network as a whole. Being all powerful, the BS can detect such malicious data or code using software that is able to detect and remove such malicious content, or any foreseen threat.

Table 3.1 – Node Neighbor Activity Table at Node X

Node	p_counter	UL	DL
Node Y	15	No	Yes
Node Z	4	Yes	No
Node U	6	Yes	No
Node V	0	No	No

3.2.4. Report Accumulation Epoch

The Report Accumulation Epoch is initiated periodically so that every node informs the BS about its neighbors and the packet communication with them, if any. In this epoch, two types of reports sent by a node to the BS are differentiated; the neighbor report and the activity report.

The neighbor report has a message type of 11 and is sent by a node every time it has no DL neighbor to send its packets through. This case occurs when the node is first deployed or whenever it is isolated from the network and has no DL in a given period. In the neighbor report, the node sends a list of its neighbors to the BS.

As for the activity report, it has a message type of 12 and is sent periodically by an active node to inform the BS about its neighbor nodes and their corresponding p_counter values, which shows the number of packets sent through or forwarded from this neighbor towards the BS.

Note that the time period of sending a report, the activity period, is a network parameter chosen in the Initialization Epoch to be of the order of several magnitudes of the period of sending a normal packet containing readings to the BS.

To ensure proper receipt at the BS, each node sending a report in this epoch, through its next hop neighbor must listen to that next hop for a time t_timeout (a time chosen higher but comparable to the node's time to process and transmit a packet) in order to make sure that the latter has in fact forwarded its packet. If the next hop fails to forward the report during the timeout interval, the node broadcasts its report through all of its neighbors. Every receiving node will send the packet normally through its next hop and performs a similar action.

Note that the overhead incurred is justified for two reasons. The first is to assure that these reports, which are an essential part of CENTERA, reach the BS. The second is to help the BS locate and punish the uncooperative nodes. Also note that the broadcasting will not occur frequently in all the nodes, since bad nodes not correctly performing their jobs in sending/forwarding reports will be quickly isolated.

The nodes' neighbor report, shown in Figure 3.5, has the following format—
{Message Type = 11 [one byte], Sender ID [one byte], Message Number [one byte],

Total Neighbors [one byte], MESSAGE [(number of neighbors) bytes], MAC [x bytes]}. The message number is needed for nodes to drop multiple copies of the same packet (in case of broadcasting).

As for the nodes' activity report, also shown in Figure 3.5, it has the following format—{Message Type = 12 [one byte], Sender ID [one byte], Message Number [one byte], UL Neighbors [half a byte], Normal Neighbors [half a byte], MESSAGE [(2 + (2×number of UL neighbors) + (number of normal neighbors)) bytes], MAC [x bytes]}.

The message number is needed for nodes to drop multiple copies of the same packet (in case of broadcasting). The message starts with the DL neighbor ID followed by its corresponding p_counter; then each UL neighbor ID is followed by its corresponding p_counter; then a list of the normal [neither UL nor DL] neighbors.

Note that a node does not send normal reading packets until it receives its UL nodes and its next hop DL from the BS.

In case of an authentication scheme set, similar to the normal messages, the neighbor/activity report communicated in this epoch must be signed by its initiator to be verified in every hop until it reaches the BS. Also, every node forwarding this report encrypts the signature by its own key for its neighbor to verify that it is receiving a packet from an UL neighbor. Any wrong signature causes the packet to be dropped.

3.2.5. Node Analysis and Metric Calculations Epoch

After the BS collects and verifies the neighbor/activity reports of all the nodes, the Node Analysis and Metric Calculations Epoch is initiated. The BS saves the neighbors of each node with the respective counter values as sent by each node and performs a series of checks to detect all discrepancies and misbehaviors in the network. The BS in this epoch either flags misbehaving nodes as bad or put them on DL

probation (indicating a problem with its DL) or UL probation (indicating a problem with an UL node) or neighborhood probation (indicating a problem with a neighbor). A node is put on probation when the BS decides to give the node the benefit of the doubt and give it another chance under different circumstances (different UL or DL). When a node reaches the maximum allowed number of probations as set by the administrator, it is flagged as bad; note that the maximum number of probations is a parameter set by the administrator in the Initialization Epoch based on the network environment and the sensitivity of the exchanged data. Moreover, if any node is considered bad for any reason, *i.e.*, falsely manipulating counters or sending illogical data or reached the limit of probations, the BS neglects its report and counters in its calculations and checks.

The BS checks proceed as follows. First, the BS verifies neighborhoods of each node. It keeps tracks of nodes removing and then adding their neighbors and flags them as bad after a specific number of unexplained changes. A neighborhood relationship is considered good if it is confirmed by the two neighboring nodes.

After that, the BS validates the reports and counter values. It flags as bad each node declaring forwarding packets through a non-DL neighbor or forwarding packets for a non-UL neighbor. Then the BS calculates the actual packets initiated by each node as the difference between its p_counter to its DL and the rest of the p_counters. If the number of initiated packets is negative, the node is directly flagged as bad.

After those checks, the BS analyzes the nodes and detects potential misbehaving nodes such as packet droppers, lying nodes, colluding nodes, *etc.* It assesses the values of all the counters by comparing them to its received packet numbers and crossing them among all neighboring nodes in the network. So, the BS calculates *Dnb* as the difference between every node's claimed number of forwarded/sent packets and what was actually received by the BS from it. Then the BS

calculates the difference *diff* between every node's claimed number of forwarded/sent packets and its DL's claimed number of forwarded packets for this node. The nodes are evaluated based on the values of *Dnb* and *diff* as summarized in Table 3.2 and detailed as follows:

- if $Dnb = 0$,
 - if $diff = 0$ the node is good.
 - if $diff \neq 0$ then increment the DL probation of the node and the UL probation of its DL. This decision is taken because *diff* is different than zero in one of three possible cases as follows: 1- DL maybe lying, 2- the node has a colluding partner down the path dropping its extra undeclared packets, or 3- the node has a clone with dual personality down the path initiating packets in its name.
- if $Dnb < 0$,
 - flag node as bad—node is lying since it is declaring less packets than what was actually received by the BS.
- if $Dnb > 0$,
 - if $diff \neq 0$ then increment the DL probation of the node and the UL probation of its DL; since node may be manipulating its counters, or its DL is dropping packets, or the link is noisy between the two nodes.
 - if $diff = 0$, node is considered as good since its DL has confirmed its declaration at its own responsibility, to be accounted for in later iterations.

Table 3.2 – Nodes Evaluation by the BS

Dnb	Diff	Result
Equals 0	Equals 0	Good Node
	Not equal to 0	Probation (Node and DL)
<0	---	Node is Bad
>0	Not equal to 0	Probation (Node and DL)
	Equals 0	Good Node – Confirmed by DL

The BS then approximates the battery life of every node based on its activity estimated by the number of received packets. The BS accounts for the number of transmitted/received packets, signed/verified packets, and encrypted/decrypted signatures. The BS calculates the different quality metrics for each packet type. Note that all the quality metrics assume values between 0 and 1. The maliciousness is calculated based on the ratio of the bad packets to the total packets received, as follows:

$$maliciousness(N) = \frac{\sum \text{bad packets received from } N}{\sum \text{packets received from } N} \quad (3.1)$$

Using information from all the good packets it received, the sink BS calculates the competence and cooperation of all the WSN nodes. The competence shows the ability of a node to properly deliver a packet to the sink BS and is calculated as the ratio of the packets received by the BS to the packets sent by the node, as follows:

$$competence(N) = \frac{\sum \text{packets received by BS from } N}{\sum \text{packets actually sent by } N} \quad (3.2)$$

where the count of packets sent by a node is calculated using its p_counters, which can be checked using the p_counters of the downlink and uplink nodes.

The cooperation shows the willingness of a node N to cooperate and forward packets sent by others towards the sink BS. Cooperation is calculated as the weighted ratio of the number of packets sent by the ULs of N through it over the total number of packets sent by those ULs, as follows:

$$\begin{aligned} & \text{cooperation}(N) \\ = & \frac{\sum_{ULs\ of\ N}(a * pkts\ rcvd\ by\ BS\ from\ UL / packets\ sent\ by\ UL)}{count\ a} \end{aligned} \quad (3.3)$$

Note that a is the weight of each uplink node (inversely) related to its maliciousness, as follows: as follows:

$$a_i = 1 - \text{maliciousness}(i) \quad (3.4)$$

The sink BS then calculates two trust values for each node: a Data Trust value and a Forwarding Trust value. The Data Trust of a node N is an indication of the benign nature of the packets of N. It is calculated based on the maliciousness of the node while taking into account the cooperation value (in order to force nodes to cooperate to increase their Data Trust). Data Trust assumes values between 0 and 1 and is calculated as follows:

$$\text{Data Trust}(N) = \frac{(1 - \text{maliciousness}(N)) * (\text{cooperation}(N))}{\text{competence}(N)} \quad (3.5)$$

The Forwarding Trust of a node is an indication of the trust in a node's ability to forward a packet and being confident that the packet will be delivered successfully to the sink BS. The Forwarding Trust is calculated based on the approximated battery level and the competence values of a node, as follows: as follows:

$$\begin{aligned}
 & \textit{ForwardingTrust}(N) = \\
 & (\textit{Approximated battery level}(N)) * (\textit{competence}(N))
 \end{aligned}
 \tag{3.6}$$

3.2.6. *Bad Nodes Isolation Epoch*

After calculating the quality metrics and trust values for all the nodes, the Bad Nodes Isolation Epoch is initiated. Any node detected as bad in the previous epoch will be isolated from the network for a number of activity periods according to its Data Trust (dtrust) level and its banNum value. The banNum is an indicator showing the bad level of the node through time. BanNum is set to one for all nodes and is incremented every time a node is detected as bad.

This epoch utilizes an effective and efficient method to isolate the detected bad node based on its history and current actions according to the following:

- if (dtrust > 0.8) then banRem = previous value of banNum
- else if (dtrust > 0.7) then banRem = 2 × banNum
- else if (dtrust > 0.6) then banRem = 3 × banNum
- else if (dtrust > 0.5) then banRem = 4 × banNum
- else if (dtrust > 0.4) then banRem = 5 × banNum
- else if (dtrust > 0.3) then banRem = 6 × banNum
- else if (dtrust > 0.2) then banRem = 7 × banNum
- else banRem = 8 × banNum

The BS increments the number of successive good periods for every active node not detected as bad nor put on probation in this epoch. When this number reaches a certain threshold (preset by the administrator in the Initialization Epoch), the BS

rewards the node by decrementing its `banNum` if it were greater than one, or decrementing its probation number otherwise.

3.2.7. Basic Routing Epoch

With the bad nodes isolated from the network, the BS starts the Basic Routing Epoch to find the shortest path for every node towards the BS. In this epoch, the BS first removes the link between neighbors put on probation to check other paths, if any, and tests the behavior of nodes and pinpoints the bad ones. Then it uses the hop cost set by the administrator in the Initialization Epoch for all the remaining links and the forwarding trust (*frust*) for each node as the weights for Dijkstra's Algorithm to find the shortest and balanced routing paths of the network. From this epoch, the BS discovers the UL neighbors of every node and the next hop DL neighbor of every node.

3.2.8. Routing Information Dissemination Epoch

Finally the Routing Information Dissemination Epoch is initiated to synchronize the pass number and distribute the routing information to the network sensor nodes. The synchronization is done by sending the number of activity periods (the pass number) as seen by the BS and thus, all nodes will be synchronized.

As for the efficient dissemination of the routing information, the epoch tries to minimize duplicate information sent to the nodes in order to minimize their communication overhead. This is done by calculating path fragments in the WSN. The path fragment is a path without any bisection. The BS determines the path fragments by first determining all the overlapping paths and then deducing the list of all the path fragments (overlapping by a maximum of one node) and finally uniquely numbering each fragment. This way the overhead of the update messages transmitted through the

sensor nodes is decreased to a minimum.

The path fragments, shown in Figure 3.5, have the following format—
{Message Type = 21 [one byte], Path Number [one byte], Path Intersection [one byte of the form path.node], Total Number [one byte], List of N Nodes [N bytes], MAC [x bytes]}.

The path number is the unique number that the BS gives to each path, and the path intersection has the format path.node specifying to which node of which path is the current path connected to. The total number specifies the number of nodes in the current path. When a node receives a path fragment, it may encounter three cases:

1. If the node finds itself to be part of the path, it saves the path number together with its location in the current path and the uplink node for that path. In addition, it performs the following:
 - a. it sets its next hop as the previous location node in the path fragment
 - b. it adds to its uplink neighbors the next location node in the path fragment
 - c. it forwards the packet to the next location node in the path fragment
2. If the node finds itself to be the intersection byte node (path.node = itself)
 - a. it adds the new path number to the paths it belongs to, together with the uplink to reach that path (in case there were more paths fragmenting from that path)
 - b. it adds to its uplink neighbors the next location node in the path fragment
 - c. it forwards the packet to the next location node in the path fragment
3. If the node is not part of the path and it is not itself the intersection node, it

checks if it has previously saved the path in the intersection node (path.node). This case occurs when a further-away path fragment is sent through the preceding distributed fragments from the BS; in other words, a closer node to the BS will not appear in the farther path fragment even though it is part of the full path:

- a. it adds the new path number to the paths it belongs to, together with the uplink to reach that path (in case there were more paths fragmenting from that path)
- b. it forwards the packet to its uplink neighbor to reach the path in the intersection node (path.node)—this uplink neighbor is saved from a previous packet

From this point on and until the next path fragment message, every node sends its periodic reading only through its next hop neighbors and forwards only the packets of its designated uplink neighbors as instructed by the BS.

Note that in case of an authentication scheme set, the BS signs each path fragment before forwarding it.

CHAPTER 4

ATTACKS AND MISBEHAVIORS

This chapter explains a list of attacks and misbehaviors that can affect the nodes in particular and the network in general, and then analyzes how CENTERA detects them and isolate their effect from the network. It will be shown that CENTERA is able to mitigate the effects of all the attacks listed in Section 1.2.2.

4.1. External Attackers

Using any authentication scheme, whether symmetric or asymmetric, the system nodes directly reject any unauthenticated packet coming from an outsider attacker node. Thus, an attacker physically penetrating the system fails to inject any packets into the network. The most harm it can do is some localized noise. Of course this is considered as a simple attacker.

It should be noted here that if the application of the WSN communicates sensitive data, the admin, at the Initialization Epoch, may choose to protect the data and force the nodes to encrypt the packet payload, at the expense of increased energy overhead. In this section, only external attackers are considered. Any external attacker that takes over a node and uses it to launch its attack is considered as an internal attacker discussed in the following sections.

4.2. Protocol Specific Attacks

There are several types of attacks or misbehaviors directly related to CENTERA and its functions. As discussed in Chapter 3, the BS in CENTERA sets and

distributes the routing paths from every node to the BS. Thus, the first type that the BS detects is the receipt of a packet on a different path than that designated. Knowing that such misbehavior requires the collusion of two nodes, the BS performs the proper checks, and distinguishes a couple of two candidate colluding nodes. The BS puts these nodes on probation and keeps them under surveillance; and isolates whichever set repeating the error.

Another type of protocol specific attack is a node sending a wrong message format; thus disregarding the rule that in a report the DL should be the first node, followed by the UL (if any) followed by the rest of the neighbors (if any). So any node sending nonconforming messages to this rule is detected as a bad node by the BS. Note that the node may be a malfunctioning node just misplacing its neighbors or a bad node deliberately changing the positions to decrease the trust of its neighbors. It may even be actually trying to send its packets not through its DL and forwarding the packets of a non-UL neighbor. In any case, this node is considered bad and negatively affecting the proper functioning of CENTERA, and thus, it is directly banned by the BS.

A similar kind of misbehavior is the manipulation of counters that results in a negative number of packets initiated by the node. As described in Chapter 3, the number of packets initiated by a node is calculated as the difference between the p_counter to the DL and the sum of the p_counters to the UL neighbors. If this difference results in a negative value, the node may be malfunctioning or deliberately manipulating its counters and should be banned from the system.

4.3. Bad Packet Attacks

This type of attack is divided into two main parts. The first is when a node is initiating malicious packets intended to harm the nodes or the BS. This type is directly

detected by the BS, as it checks all the received packets for any maliciousness. The BS definitely bans this node from the system and isolates its harmful effects.

The second type is a simple attack or misbehavior either by a malfunctioning node or a bad node intending to just flood the network with erroneous packets. Such packets may include an invalid message, an invalid signature, or an unverified signature. This type of attack is directly dropped by the neighboring nodes and thus its effect is localized and minimized.

This type of attack includes flooding DoS attacks, energy drain attack, and even packet injection attacks, among others, and the BS in CENTERA manages to detect all attacks of this type.

4.4. Packet Number Discrepancies

This type is the most prevailing type of attacks that is very easy to launch yet very effective. This attack includes packet dropping by uncooperative nodes, nodes lying about their counters, incompetent nodes due to malfunctions or noisy environments. Specifically, as per section 1.2.2 definitions, this type includes selfishness attacks, black-hole, and gray-hole.

As discussed in Chapter 3, the BS always compares the number of packets received from a node to the number declared by this node; also it compares the number of packets declared to be forwarded by a node to the number declared to be forwarded by its DL. From these comparisons, the BS locates the problem in a link between two nodes; however, it can't specify exactly whether a node is lying or its neighbor is dropping or if there is noise on the link. So, the BS gives these nodes the benefit of the doubt and provides them another chance after removing the link between them. Then, based on a parameter set by the administrator, a node is isolated when the number of

maximum allowed probations is reached.

4.5. Broadcasting Nodes

Another type of attack is an attempt to disrupt the network by always broadcasting packets to all its neighbors. This type is detected and isolated in two stages. First, the effect of the broadcast is locally removed directly since all of its non DL neighbors drop this packet. As for the high transmission rate of the node, this is detected by the BS and decides to put the node on probation or directly isolate it depending on the packet transmission overhead.

Broadcasting nodes may include the same type of attacks as presented in the Bad Packet attack depending on the exact method an attack is performed. Also here the BS is able to detect all the presented attacks.

4.6. Colluding Nodes

This type of attacks is somehow advanced, where two nodes are colluding to disrupt the network or bias it to their advantage. An example of colluding nodes that can really impair the proper operation of the system is the case where an upstream node A is colluding with a downstream node B to drop its extra undeclared packets. This attack aims at decreasing the trust value of benign nodes in the network and banning them. Attacker A sends more packets than it later declares in its activity report, and its benign DL forwards all of its packets; however after some hops down the path, node B drops the extra packets (upon previous agreement) from source A. The goal is to trick the BS into flagging the DL node of A as a lying node.

In CENTERA, after the BS detects the difference in the packets sent by A and those forwarded by its DL, and the difference between the packets sent by the UL of B

and those forwarded by B, the BS puts both pairs on probation and changes the links between them. In the following periods, the colluding nodes persist in the same attempt to disrupt the network, while the other nodes continue operating normally. So, after the number of probations of A and B reaches the maximum allowed, the BS flags them as bad nodes and isolates them from the network. Note that as the number of colluding nodes increases, the BS is faced with more and more misleading reports, until a limit where the logic fails and the BS's decisions start to be inaccurate and erroneous; *i.e.*, the system fails.

4.7. Node ID Attacks

This type includes node replication attack, Sybil attacks, ID spoofing attacks, and even whitewashing attacks. In node replication attack, the attacker introduces replicas of one compromised node using its same ID at different locations of the network. Upon the introduction of the replica into a new neighborhood—*i.e.*, connecting to a set of different nodes than the original one, the system may be encountered with two cases. The first is the case where the replica directly starts sending packets without proper introduction with the hello messages, the neighbors will reject its packets as it is assigned as neither their DL nor as their UL by the BS; and thus the attack is directly isolated in this case.

The second case occurs if the added replica starts with a proper introduction of hello messages, then, the neighboring nodes accept it and send their updated activity report with the replica as a neighbor. Here, there are two cases: 1- if the BS receives two different activity reports containing different neighbors from the same node ID, it directly detects the replication and isolates this node ID, the two replicas, from the network; 2- If the replicas are more sophisticated and sending a unified activity report

with neighbors from both neighborhoods, the BS detects a neighborhood error (to be discussed in the following section) and the node ID will be isolated from the network.

Sybil attacks occur when an attacker uses several invented or stolen IDs to sign packets using their IDs and encrypt these signatures by its own ID. This way the attacker injects packets in the name of another node after encrypting its signature by its own ID to appear as a legitimate packet flowing through this path of the network.

Note that with the incorporation of the identity-based authentication scheme, the attacker cannot affect the network without acquiring the master key, which is saved offline away from the network, or having access to one or more nodes. Regarding acquiring the master key, it is considered as highly improbable due to the fact that it is saved offline by the administrator. As for the control over nodes, the attacker is considered as a replica with a dual ID. The analysis is similar to the one presented above; the BS detects differences in the number of forwarded and sent packets, puts nodes on probation, changes paths, and detects and isolates the bad nodes.

Similar reasoning can be done to show that the BS in CENTERA can easily detect ID spoofing attacks, whether the attacker is powerful and has the knowledge of the private key of another node or not.

Whitewashing attacks are also considered in this family of attacks, as the attacker is attempting to change its ID to avoid punishment; however, as the attacker does not have the network master key, it cannot create itself a new legitimate identity acceptable by the network nodes.

4.8. False Neighborhood Attacks

This type of attack includes asymmetric neighborhoods in nodes or colluding nodes adding false neighborhood to produce a wormhole for example. The first part is

when a node A is claiming to be neighbors with node B, and node B is not. This type of misbehavior may be caused by a malfunction or a bad nature. In both cases, the BS puts nodes on as many probations as the number of such differences and discrepancies. Thus depending on the majority, the BS is able to detect and isolate such bad nodes.

As for the colluding nodes adding false neighborhood between them, if the nodes are able to forward packets between them in any way, then there is actually a link and neighborhood between them. So, they are evaluated normally in the system depending on their behaviors. On the other hand, if those nodes are unable to forward packets between them, the BS detects the dropped packets between them, when they are associated as UL—DL neighbors. Thus, in any case the BS detects neighborhood attacks without being able to confirm their actual positions. For improved accuracy and localization of such misbehaviors, the administrator may decide to use secure positioning in order to geographically locate nodes and better validate neighborhoods.

4.9. Bad Mouthing and False Praising Attacks

As our system does not explicitly have reputations given by one node to another, this attack can be implicitly launched by manipulating counters, and thus attempting to affect the decisions taken by the BS. However this attack can be detected easily by the BS as described in Section 4.4.

4.10. Transient Behavior Attacks

Whether being normal on-off or conflicting behavior, or even an intelligent behavior, this attack is very common and usually effective in allowing a node to get through with its bad deeds. However, in CENTERA, as described in Section 3.2,

misbehaviors are not directly forgotten, on the contrary, their effect remains based on an administrator preset parameter. Thus, an attacker remains to be considered bad and needs some time to improve its trust values and prove itself as a benign node again.

4.11. Sinkhole Attacks

As the BS is the sole entity responsible for routing information, this attack is inherently avoided, as a sensor node does not have the ability to draw traffic to itself by advertising fake routing information.

4.12. Modification Attacks

Using authentication techniques in CENTERA provides data integrity, and thus any modification of a packet by any node other than its initiator is directly detected and the packet is dropped. As a result the BS detects the counter differences and isolates the bad node.

4.13. Packet Delay and Packet Replay Attacks

These two attacks are considered harmful whenever they target report packets, which constitute an essential epoch in CENTERA.

Regarding the packet delay attack, its effect is mitigated in CENTERA by the implemented mechanism to ensure the proper receipt of report messages at the BS. Thus a node is obliged to listen to that next hop for a time in order to make sure that the latter has in fact forwarded its packet, else the node broadcasts the report through all of its neighbors.

As for the packet replay attack, it will have no effect on CENTERA as the

report message contains a message number to ensure the freshness of the report and to prevent its retransmissions.

Table 4.1 summarizes all of the discussed attacks in this Chapter with a brief description of each and how it is mitigated by CENTERA.

Table 4.1 – Summary of Attacks

Attack	Description	Mitigation
External Attacks		
Outside the system	Perform any hostile action	- Rejected by authentication - Encryption if required
Protocol Specific Attacks		
Related to CENTERA operations	- Packet Receipt on wrong path - Wrong message format - Negative initiated packets	BS detects and bans bad node
Bad Packets Attacks		
Bad transmitted packets	1. Malicious packets 2. Invalid message/signature	1. Detected by BS 2. Localized Effect
Broadcasting Attacks		
Broadcasting good packets	- Malfunctioning or bad	- Dropped locally by non DL - Banned by BS
Packet Number Discrepancies Attacks		
Wrong counter values	- Uncooperative Nodes - Incompetent Nodes - Lying about counters - Noisy environments - Selfish, black and grey holes	1. BS locates problem 2. Puts two nodes on probation 3. Bans bad if persists
Colluding Attacks		
Strong – Disrupt the network or bias it to attackers' advantage	Colluding with downstream node to drop its extra undeclared packets to decrease trust of benign nodes	1. BS detects two pairs of nodes (probation) 2. Banning
Node ID Attacks		
Changing or Creating Node ID	1. Replication 2. Sybil (not taking over a node) 3. ID Spoofing 4. Whitewashing	1. Locally or BS 2. Not possible with IBE 3. Not Possible with IBE 4. Not possible with IBE

False Neighborhood Attacks		
Wrong Neighborhood Information	1. Asymmetric neighborhoods 2. Colluding nodes (wormholes)	1. Probation by BS 2. Either evaluated normally BS detects dropped packets Improved by Secure Positioning
Bad Mouting or False Praising Attacks		
Inherently blocked as reputations are not exchanged between nodes		
Transient Behavior Attacks		
	- On/Off - Conflicting Behavior - Intelligent Behavior	Bad node needs time to clear its bad reputation
Sinkhole Attacks		
Node draws traffic to itself by advertising fake routing information	Inherently targeted since only the BS is responsible for routing information	
Modification Attacks		
Blocked by data integrity enforced by Authentication techniques		
Packet Delay and Packet Replay Attacks		
1. Packet Delay mitigated for Reports 2. Packet Replay have no effect due to message number in Reports to ensure freshness		

CHAPTER 5

SIMULATION VERIFICATION

5.1. Simulation Setup

In order to evaluate CENTERA and prove its correctness in providing routing information while creating a trusted environment and isolating bad nodes, we have used the TOSSIM simulator to simulate a grid of Micaz sensors running TinyOS [49]. We used different topologies and different network sizes, a linear network of 30 nodes, a tree of 40 nodes (where each node has three daughters), and a grid network of 5×5 , 9×9 , 15×15 , and 31×31 nodes.

As for the authentication technique, using [33], we choose the asymmetric key cipher technique PBC, which was found to be the best authentication technique to be incorporated into CENTERA in a hostile environment. We used the TinyPairing library [50,51] and modified the revised BLS-SS (Boneh, Lynn, and Shacham - Short Signatures) and the revised BF-IBE (Boneh and Franklin – Identity Based Encryption) to have the short signatures based on IBE (IBE-SS) to sign a message using the private key and an IBE encryption to encrypt other node's signature using the private key, where the public key (ID) is used to verify and decrypt respectively.

This way each receiving node can be sure that the source node is indeed the true sender of the packet and the BS can calculate the trust values for the nodes, and properly construct routing paths and detect and isolate malicious/malfunctioning nodes. The attacker model in this case can be assumed to be strong, with the power to take over a node and use it to send packets. With PBC, CENTERA can detect the malicious or compromised node and isolate it completely from the network, thus increasing the

network lifetime.

In the network tested, we varied the hop cost from 0, 0.25, 0.5, up to 1 and tested the correctness of our protocol and the energy overhead imposed by the additional transmission/receipt and the cryptographic calculations namely signing/verifying and encryption/decryption.

We also assigned several bad nodes to see the effectiveness of our protocol as follows: Node 12 partially non-cooperative dropping one out of every three packets forwarded through it; node 7 is an outsider node not belonging to the system; node 23 is a partially malicious node sending one bad packet every three packets it sends; node 17 is declaring sending packets through a non-DL neighbor; node 3 is incrementing its DL counters; and node 19 is incrementing one of its UL counters.

5.2. Simulation Results

In this section, we show the results of the different topologies, sizes, parameters, and attacks we implemented.

5.2.1. Different Topologies and Sizes

5.2.1.1. The Linear Topology



Fig. 5.1 – The Linear Topology

In the linear network, we simulated 30 nodes with the sink BS as the first node. In this type of topology, shown in Figure 5.1, each non-border node has only two

neighbors and thus the hop cost and weights do not give any difference in the routing path to reach the BS. Also, precautions should be taken in the availability of bad nodes; since any isolation of a non-border node causes a network partition. In this case specifically and whenever any node has only one DL node to the BS in general, banning a bad node should be based on the type of misbehavior (set by the administrator in the Initialization Epoch). In other words, a tradeoff should be made between the misbehavior's effect and the importance of the lost partition from the network. So if the bad node is sending malicious packets for example rendering the whole network useless the node should be banned at the cost of discarding a part of the network; however for simple cooperation errors, the administrator may choose to tolerate this misbehavior for the benefit of keeping the network alive.

5.2.1.2. The Tree Topology

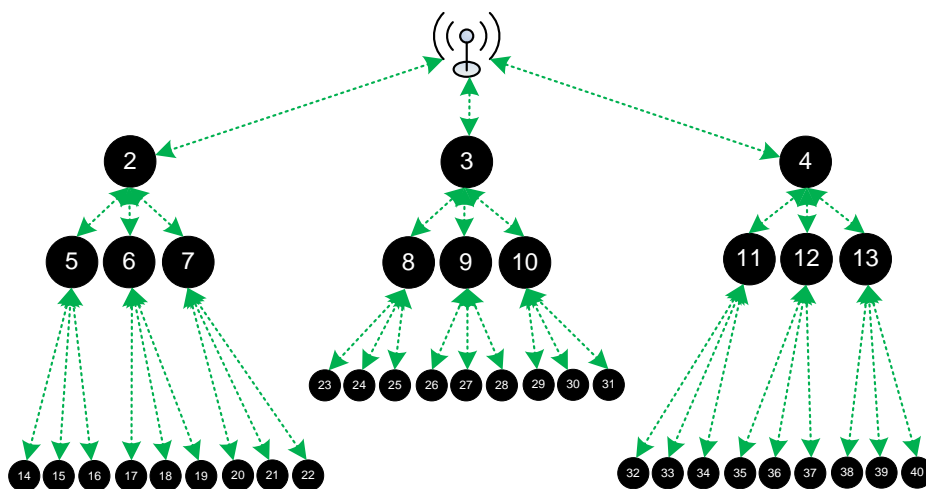


Fig. 5.2 – The Tree Topology

Another special type of topology is the tree topology, shown in Figure 5.2. In the tree topology, each node also has only one path the sink BS, and thus similar

reasoning is done as the linear topology case.

The hop cost does not change the routing path and a tradeoff should be taken as when to isolate a node and when to bare its misbehavior.

5.2.1.3. The Grid Topology

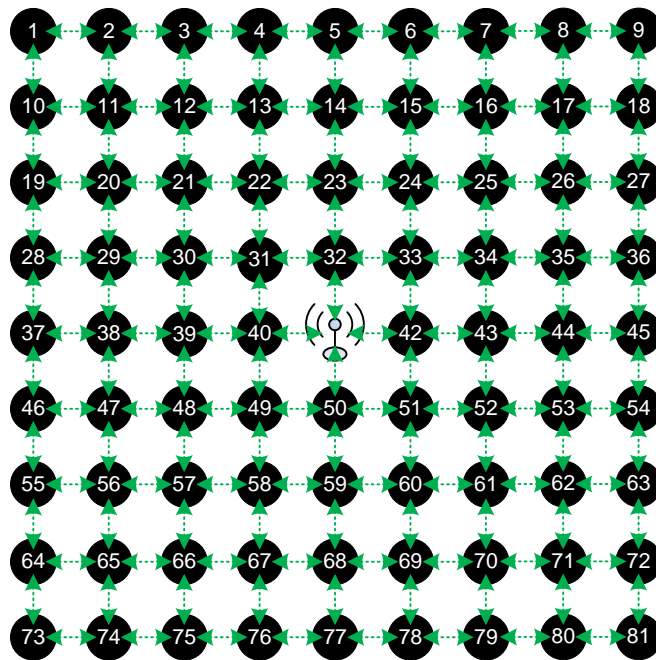


Fig. 5.3 – The Grid Topology (9x9)

We simulate a general topology, the grid network for different number of nodes ranging from a simple 5×5 network to 9×9 , 15×15 , and a large 31×31 network. In all those topologies, the sink BS is set as the center node and there exists a connection between any two adjacent nodes. We show an example of a 9×9 grid topology in Figure 5.3, where the green arrows show radio range between the nodes. Initially, the different epochs of CENETRA execute correctly and the BS is able to

create shortest paths routing information and distribute such information to the different nodes.

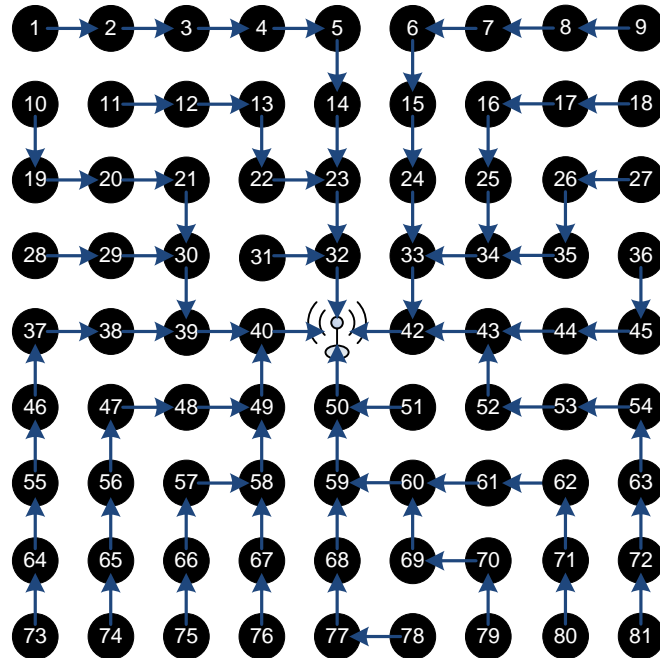


Fig. 5.4 – Initial Routing Paths in the 9x9 Grid Topology (Hop Cost = 1)

Figure 5.4 shows the network initial routing paths of the different network nodes with hop cost equal to 1. Note that as time passes and the battery lives of nodes depreciate, the paths change to distribute the load in a balanced way and depreciate all nodes equally.

Changing the hop cost in Dijkstra’s algorithm from one to zero gives the possibility to add longer paths from nodes. For hop cost equal to one, the BS chooses the most efficient path from the set of shortest paths for each node, while keeping the network relatively balanced. For hop cost equal to zero, the choice depends solely on the “ftrust” value (incorporating the battery life and competence) totally neglecting the hop count in the decision. Thus, the BS chooses the path with the least ftrust at the

expense of increasing the overall communication energy overhead in the network as a whole. Also, with hop cost equal to zero, the routing paths are very dynamic and highly changing every period.

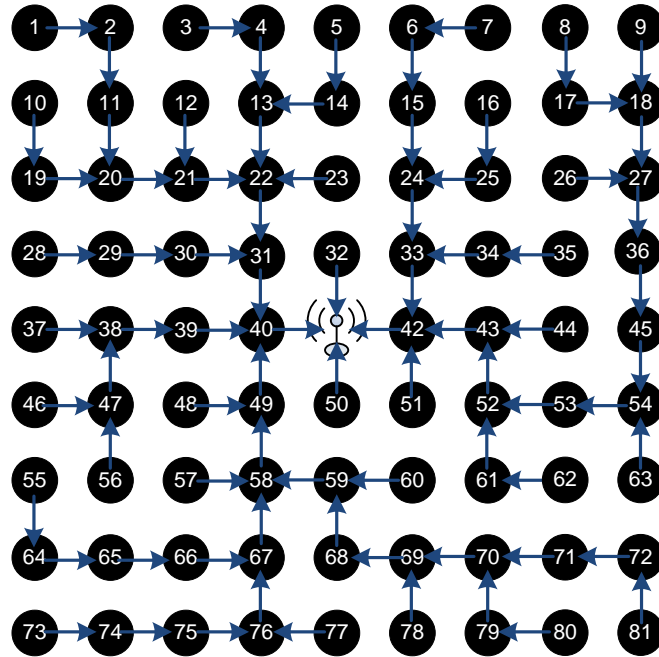


Fig. 5.5 – Routing Paths with hop cost equals zero

Figure 5.5 shows the routing paths of the 9×9 network with hop cost equal to zero. In this pass, the BS estimated that battery levels of nodes 32 and 50 are very low, of node 42 is 60%, and of node 40 is 70%. It is obvious how the protocol almost fully depleted two nodes while the other two nodes are still good on power. Also the paths are very long and most nodes depend on one BS neighbor (50 nodes are forwarding their packets through node 40!) causing a bottleneck, fast depletion and a higher probability of dropped packets. Consequently, in our tested grid topology, removing the hop cost and depending solely on the frust is not a feasible option to consider.

Figure 5.6 shows the routing paths with hop cost equal to 0.25 (b) and 0.5 (a).

Similar results and analysis are seen that as the hop cost decreases, the routing paths are longer and more dynamic. It is clear that as the hop cost increases the paths are more symmetrical and balanced.

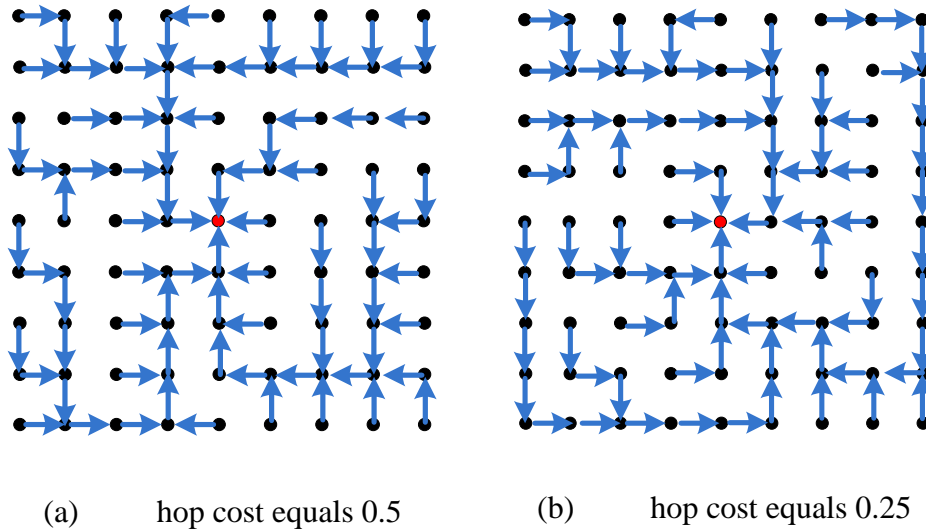


Fig. 5.6 – Routing Paths

Table 5.1 shows the load on the BS neighbors in this specific period and assures our conclusion that as the cost of the hop increases the routing paths are more balanced and less dynamic every period. It should be noted that longer and unbalanced paths have the advantage of depleting the whole network together, which could be useful in some types of irregular networks. This comes at the expense of increasing the overall network communication energy consumption and longer paths, which by itself could pose the risk of higher packet dropping rate. Similar results are found in smaller topologies such as 5×5 and larger topologies such as the 15×15 grid.

As for the 31×31 topology, it contains 961 nodes and thus requires more than one byte to account for the node IDs. Accordingly, we increased the sizes of the affected fields in the packets' headers, namely the fields containing the IDs, and

simulated the network again. The protocol proved to be correct even for such large networks, giving a balanced and shortest path routing for all nodes in the network.

Table 5.1 – Load on the BS neighbors with respect to hop cost

Hop Cost	Node 32	Node 40	Node 42	Node 50
1	12	26	25	13
0.5	6	29	0	41
0.25	1	0	30	45
0	0	50	26	0

5.2.2. Different Attacks and Misbehaviors

Following we discuss the detection and isolation of bad misbehaving nodes for which we assigned several bad nodes as discussed above.

5.2.2.1. Uncooperative Node

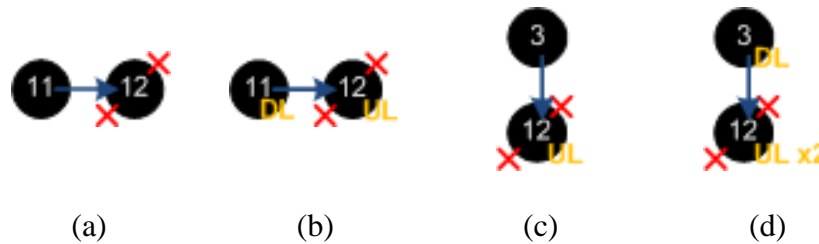


Fig. 5.7 – Uncooperative Node 12

First, node 12, as shown in Figure 5.7 (a), was set to be a partially non-cooperative node dropping one out of every three packets forwarded through it. In the sixth pass shown in Figure 5.7 (b), that is the first activity period after the BS distributed the DL and UL information to all the nodes, node 12 is put on UL probation and its UL node 11 is put on DL probation. Then the BS changes the link between the

two nodes to detect which node is misbehaving and set node 3 as an UL of node 12, shown in Figure 5.7 (c). In the following activity report period, pass 12, the BS found a difference in the counters of node 3 and node 12 and gives node 3 a DL probation and node 12 a second UL probation as shown in Figure 5.7(d). Thus With the maximum allowed probation set to 1, node 12 is set as a bad node, and thus isolated from the network for the next three periods, since its dtrust value is 0.67 and its banNum is still 1, so as per the banning system, the banRem is set to three activity periods and the banNum is incremented to 2.

Thus, as described earlier, when the BS detects a difference in what nodes are claiming to have sent/forwarded through/for each other, it gives them another chance and changes the link between them, since this may be due to a noisy link, lying node, or uncooperative node. As seen in this example, even node 12 which partially drops packets can be detected by the BS, if it persists on its bad actions. Note that decreasing the probation limit increases the decision to ban a node at the expense of false positives. On the other hand, increasing the probation limit gives the misbehaving node more time to exploit the network and drop its neighbors' packet.

5.2.2.2. Outsider Attacker and a Malicious Node

Then node 7 is an outsider attacker node not belonging to the system and node 23 is a partially malicious node sending one bad packet every three packets. The results show that node 7 is isolated from the system as it does not have the required key to sign its packets, and thus all of its packets are dropped by its neighbors 6, 8, and 16. As for node 23, the BS detects each malicious packet and set it as a bad node and then isolates it for 2 periods, since its dtrust value is 0.75 (sending only one bad packet out of four in the first period). Definitely, in subsequent periods, node 23 is further isolated for every

additional malicious packet sent.

Table 5.2 details the different values and describes the banning process. In the first activity period, the BS has just received the neighbor report from the nodes, and thus there is still no data to assess the nodes. In the second activity period, the BS has received five packets from the source node 23 out of which one is detected as a malicious packet with harmful content; so directly the BS flags node 23 as bad without any probation. Then the BS calculates the rest of its traits and values, resulting in banning node 23 for two periods and increasing its banNum to 2, as shown in the table. Note how banNum acts as a history for bad activity. In the end of the fourth period, the isolation time has ended and the BS includes the node into the network again. However, in the fifth activity period, the BS detects a malicious packet again, and thus the BS decides now to isolate the node for three periods. Note that, since node 23 is a partially malicious node and it is sending a small percentage of bad packets, its dtrust is not very low, and thus its banning period is increasing slowly over time. This would have been much more aggressive had the node been totally malicious.

Table 5.2 – Different Values of node 23 at BS every period

Activity Period	1 st	2 nd	3 rd	4 th	5 th
DL	-	32	-	-	32
Received at sink	-	5	-	-	4
Node received	-	45	-	-	92
Node forwarded	-	50	-	-	96
Bad Received	-	1	-	-	
Maliciousness	-	0.2	-	-	0.2
Competence	-	1	-	-	1
Cooperation	-	1	-	-	1
Ftrust	-	0.3	-	-	0.3
Dtrust	-	0.8	-	-	0.8
BanRem	0	2	1	0	3
BanNum	1	2	2	2	3
Probation	0	0	0	0	0
Bad	0	1	0	0	1

This example shows two main features of our system, 1- the network inherently isolates outsider nodes using the strong yet efficient authentication scheme, and 2- the BS directly recognizes the bad, misbehaving node by detecting its sent malicious content even if at a low rate. In this example, the BS isolates node 23 for two periods only as an initial countermeasure since node 23 has not had any previous bad actions. After the banning period ends, BS tests node 23 again, however this time node 23 has a history and thus when node 23 repeats its malicious activity, it is banned for five periods. This continues by increasing the banning periods before rechecking the node by giving it an additional chance; and anytime the node stops its bad deeds, its banNum starts decreasing until it is considered as a good node again.

5.2.2.3. Counter Manipulating Nodes

In the following, we show the effect of misbehavior constituting counter manipulations. Node 17 is claiming to send packets through a non-DL neighbor and thus, the BS directly detects it as a bad node for that. As the dtrust of node 17 is still equal to 1 and it has no history of bad actions, it is isolated for one period initially, the number which increases as the misbehavior persists. Note that the reason behind this misbehavior may be a malfunctioning node or a bad node trying to delude the BS into considering some good node as bad; however, in any case, this type of misbehavior should be directly stopped as it affects the correct operation of the system.

Then, we add two misbehaving nodes, node 3 incrementing its DL counters and node 19 incrementing one of its UL counters. At pass 6, the BS detects the discrepancies and puts node 3 on DL-probation and its DL, node 4, on UL-probation; it sets the DL of node 3 as node 12. As for node 19 it has no ULs for this period and it did not do anything wrong so far. In pass 12, node 3 is detected as bad since it deserved a

second DL probation. In this pass, node 19 manipulated its counters for its UL (node 10), and thus node 19 is put on UL probation and node 10 DL probation. In pass 18, node 19 is isolated. We also got similar results when repeating the simulation with node 3 decrementing its DL counters and node 19 decrementing one of its UL counters.

Thus, CENTERA can detect any node trying to manipulate its counters due to malfunctioning or due to the intention to hurt other nodes and cause them to be banned. In either case, the BS can after some checks and analysis isolate the exact misbehaving node.

5.2.2.4. Impersonator Node

To further analyze the benefits of authentication in CENTERA, we take a look at two attackers on the network, nodes A and 4. It is directly noticed that the outside attackers are isolated completely from the network in both cases. The strong attacker took over node 4. It tried to impersonate other nodes, but this is impossible without the private key that represents the identity of the node. So, the attacker started using node 4 to send bad packets into the network. The BS updated the routing paths of the nodes such that node 4 is totally isolated from the system. As for the attacker node A, without proper authentication, it is directly neglected by the all the nodes in the system, and if it used the same authentication technique, the BS directly updates the routing paths to neglect this outsider.

5.2.2.5. Broadcasting Node

We finally test a broadcasting node, node 23 that is trying to broadcast packets through nodes 14, 22, 24, and 32, either due to a malfunction or in order to disrupt the whole network. However, it is clear, from the snippet of the topology shown in Figure

5.8, how CENTERA forced nodes 14, 22, and 24 to drop the packets from 23 because they are not the DL of node 23 as indicated by the BS. Only node 32 is forwarding the packets of node 23. Thus here we can directly see the first benefit of CENTERA in preventing broadcast storms that can increase the noise levels in the network and affect the network functionality and lifetime. Also for any discrepancies declared in its p_counters, node 23 is punished and isolated as the cases stated above.

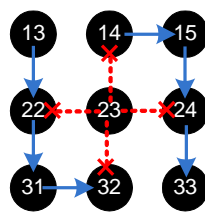


Fig 5.8 – The isolation of the broadcasting node 23

Thus, as deduced from the simulations, after the first time period, each node starts acting per its nature. This directly gets reflected in the routing paths that now avoid the bad nodes. The routing paths are updated to avoid the “bad” nodes and pass only through “good” nodes. So, all of the bad nodes are isolated and no other node is forwarding their packets.

5.2.2.6. A Comprehensive Case

In the network of Figure 5.9, we assigned several bad nodes as follows: **Node 5** is a totally malicious node that always sends bad packets towards the BS, while **node 12** is an inside attacker that is partially malicious, and sends one bad packet after every three packets it sends. Also, **node 34** is a totally non-cooperative node dropping all the packets forwarded through it, and **node 58** is partially non-cooperative dropping one out

of every three packets forwarded through it. **Node 37** is a node broadcasting packets through all of its neighbors. In addition, we introduced **node 61** as partially uncooperative and partially malicious, and **node 71** as totally uncooperative and totally malicious. Finally, there is an outside attacker who implanted a new node “A” between the nodes 16, 17, 25, and 26 to try to become part of the WSN.

Initially and similar to the normal case, the different epochs of CENTERA execute correctly and the BS is able to create shortest path routing information and distribute such information to the different nodes, as shown in Figure 5.9.

After the first time period, each node starts acting per its nature. This directly gets reflected in the routing paths that now avoid the bad nodes. It is clear in Figure 5.10 how the routing paths are updated to avoid the bad nodes and pass only through good nodes. So, all of the bad nodes are isolated from the network such that other nodes neither forward their packets nor forward packets to them.

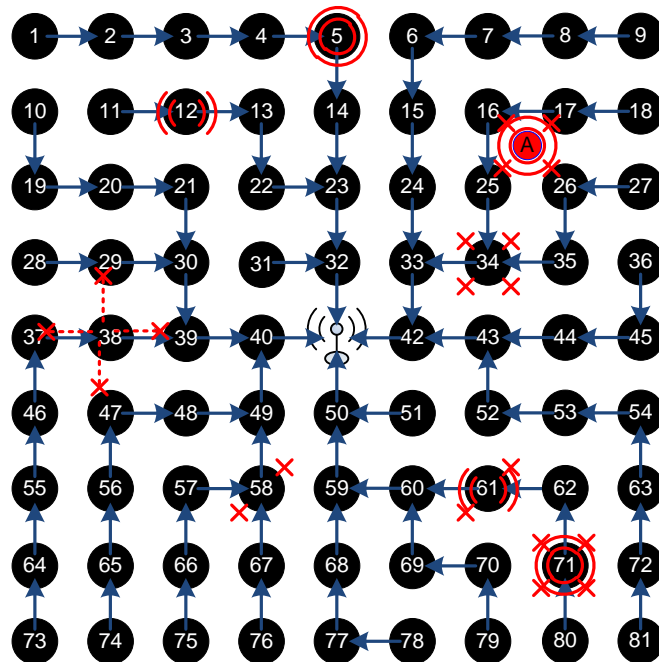


Fig. 5.9 – The Initial Routing Paths with the Attackers

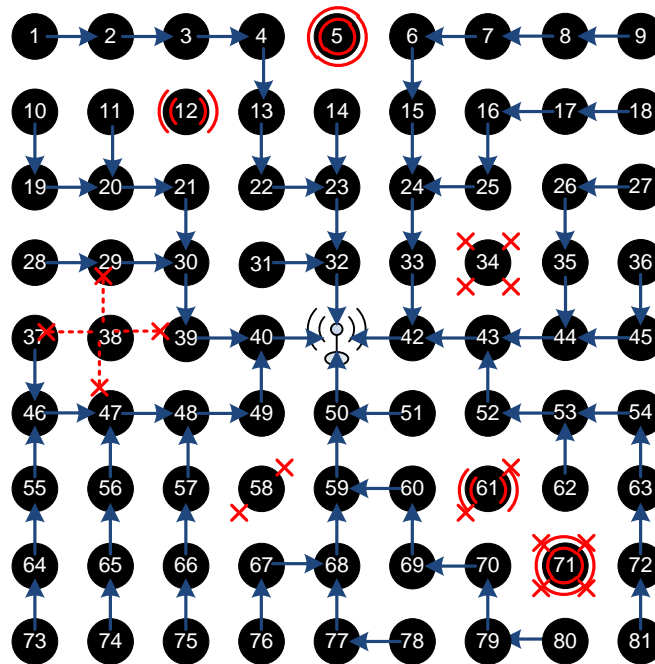


Fig. 5.10 – The Updated Routing Paths

5.3. Authentication Benefits

In order to assure complete functionality and proper usability of CENTERA, a strong yet efficient and lightweight cryptographic technique should be introduced to provide authentication of each node to the others nodes. So, we simulate CENTERA using the first two authentication techniques discussed in Chapter 3, namely the symmetric cipher RC5 and asymmetric technique PBC, to study the effect of these two authentication techniques.

5.3.1. Symmetric Cipher – RC5

For the first authentication technique, the symmetric cipher RC5 may be incorporated in any case where all nodes and even attackers are assumed to be sincere in introducing themselves in their packets and do not impersonate other nodes. The attacker model in this case is assumed to be weak where it can only send “bogus”

messages into the network; however it is not able to take over a node. Logically, by taking over a node, an attacker uses such node and impersonates others to conduct attacks.

Hence, this technique can be used in a safe environment under close administration where a bad node is merely a malfunctioning node or a weak outsider trying to join the network. All nodes are assumed to be sincere in introducing themselves in their sent packets, so that the trust model and the base station can calculate trust for the nodes and detect the malfunctioning node.

With these assumptions, the results are perfect and similar to the one shown for the next case as long as all the nodes and attackers are assumed to be *unable to change their identification or impersonate other nodes*. Otherwise, the base station will wrongly isolate a good node and thus just one strong attacker can control the whole network and render it useless.

5.3.2. Asymmetric Cipher – PBC

As for the asymmetric key cipher techniques, PBC is the best authentication technique to be incorporated into CENTERA. This way each receiving node can be sure that the source node is indeed the true sender of the packet and the BS can calculate the trust values for the nodes, and properly construct routing paths and detect and isolate malicious/malfunctioning nodes. The attacker model in this case can be assumed to be strong, with the power to take over a node and use it to send packets. With PBC, CENTERA can detect the malicious or compromised node and isolate it completely from the network, thus increasing the network lifetime. Note that in this scenario, we assume *node 4* to be compromised (taken over) by a strong attacker with the ability to abuse the node and impersonate others.

To analyze the benefits of authentication in CENTERA, we take a look at the nodes *A* and *4*. It can be directly noticed from Figure 5.11 that the outside attackers are isolated completely from the network in both cases. The strong attacker took over *node 4*. It tried to impersonate other nodes, but this is impossible without the private key that is the identity of the node. So, the attacker started using *node 4* to send bad packets into the network. The BS updated the routing paths of the nodes such that *node 4* is totally isolated from the system.

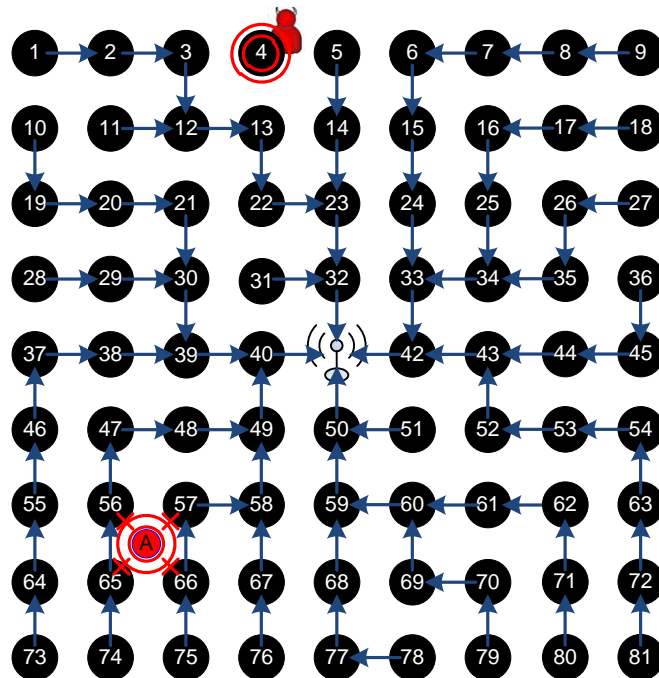


Fig. 5.11 – The Updated Routing Paths and Node Isolations

As for the attacker *node A*, without proper authentication, it is directly neglected by the all the nodes in the system, and if it used the same authentication technique, the BS directly updates the routing paths to neglect this outsider.

5.3.3. The Hybrid Techniques

The hybrid technique may also be used for CENTERA offering less security than the previous case, based on the refresh period, but leading to better energy consumption and extended network lifetime.

The decision on which authentication technique to use depends on the environment of the network, the attacker model, and the sensitivity of the data being sensed and gathered at the BS.

CHAPTER 6

ENERGY CALCULATIONS

In order to evaluate the energy consumption of CENTERA, we simulated an average sized grid topology of 9×9 nodes and extracted the bytes transmitted/received and those used under cryptographic calculations. In both cases we differentiate the initial activity period—where the nodes are not yet informed about their DL neighbors, and the remaining subsequent periods. We show the total bytes in the whole network, the average bytes per node, and the worst case in each activity period (the node that endured the maximum energy consumption).

Note that in our simulation, we choose the activity period to be 6—that is the nodes send their activity reports and the BS updates the routing paths every five periods of sending a normal message.

6.1. Communication Overhead without Authentication

Table 6.1 shows the number of bytes transmitted and received by each epoch of our system running without any authentication technique. We show the numbers in four passes where activity/neighbor reports are sent, representing the full network, the average bytes per node, and the worst case.

From Table 6.1, it can be seen that the hello messages occurred in passes 2 and 20, which is normal since pass 2 is the initial period where nodes send hello and hello reply messages to get acquainted, whereas pass 20 is the third activity period and in our simulations this is the time to send hello keep-alive messages. The network as a whole has communicated 1328 bytes of those messages, averaging to 16.6 bytes per node; and

the worst case was node 42, which actually received all of the hello replies from all of its neighbors. Note that some nodes may not receive the total number of hello replies due to noise or collisions, but CENTERA’s hello scheme is resilient to such effect. As for pass 20, the network communicated a less number of 728 bytes due to the fact that hello keep-alive messages don’t require a reply. The average was 10 bytes per node, which is logical since every node has to broadcast one keep-alive message and receive as many keep alive messages as the number of its neighbors; and each message contained just two bytes.

Table 6.1 – Bytes Transmitted and Received without Authentication.

Pass		Hello	Reports	SUB	Total OH	Normal	TOTAL
2	Network	1,328.00	130,514.00	3,510.00	135,352.00	0.00	135,352.00
	Av. Per Node	16.60	1,631.43	43.88	1,691.90	0.00	1,691.90
	Worst Case (42)	20.00	1,823.00	186.00	2,029.00	0.00	2,029.00
8	Network	0.00	6,856.00	3,574.00	10,430.00	71,680.00	82,110.00
	Av. Per Node	0.00	85.70	44.68	130.38	896.00	1,026.38
	Worst Case (32)	0.00	516.00	209.00	725.00	5,264.00	5,989.00
14	Network	0.00	6,856.00	3,479.00	10,335.00	71,680.00	82,015.00
	Av. Per Node	0.00	85.70	43.49	129.19	896.00	1,025.19
	Worst Case (50)	0.00	784.00	113.00	897.00	7,952.00	8,849.00
20	Network	728.00	6,856.00	3,916.00	11,500.00	53,760.00	65,260.00
	Av. Per Node	9.10	85.70	48.95	143.75	672.00	815.75
	Worst Case (32)	10.00	564.00	240.00	814.00	4,284.00	5,098.00

As for the reports, in the initial phase the reports consist of neighbor reports only. The network communicated 130,514 bytes averaging around 1631 bytes per node; with the worst case being 1823 bytes. This number of bytes may seem high, however this occurs only in the initial phase where the nodes still do not have downlinks and, thus, they broadcast their neighbor reports to reach the BS. So, it is considered as a startup overhead that is insignificant in the life time of the network.

In subsequent phases, the reports consist of the activity report, larger in size, but smaller communication overhead, since nodes forward them through their DLs to reach the BS. In these phases the average node communicated as low as 85 bytes of activity reports, with the worst case being 564 bytes for node 42, which is normal for a direct neighbor of the BS.

The subpaths sent by the BS to disseminate the routing paths and provide every node with its DL and ULs, show close number of communicated bytes in both phases, initial and subsequent. Averaging around 45 bytes per node, the overhead is minimal for one of the main steps in the system. Therefore the total communication overhead per node from CENTERA's epochs, when authentication is not added, reached 1691.9 communicated bytes in the initial phase, and settled at around 130–140 communicated bytes per node in subsequent phases.

Considering that the normal data packet containing sensor readings consists of 28 bytes (as set by TinyOS), the communication overhead of CENTERA ranges from 12% in normal periods to 17% in periods where the keep-alive message is sent. Note that these percentages are calculated when the activity period is taken to be as low as 6. For relatively stable networks the activity period may be chosen by the administrator to be much larger than that at for example 50, decreasing the communication overhead to less than 2%.

Figure 6.1 shows the different percentages of the communication energy dissipated by the epochs of CENTERA and the normal data packets exchanged. It is obvious that, with the exception of the first activity period, CENTERA is adding little overhead (12% to 17%) to the normal function of the WSN.

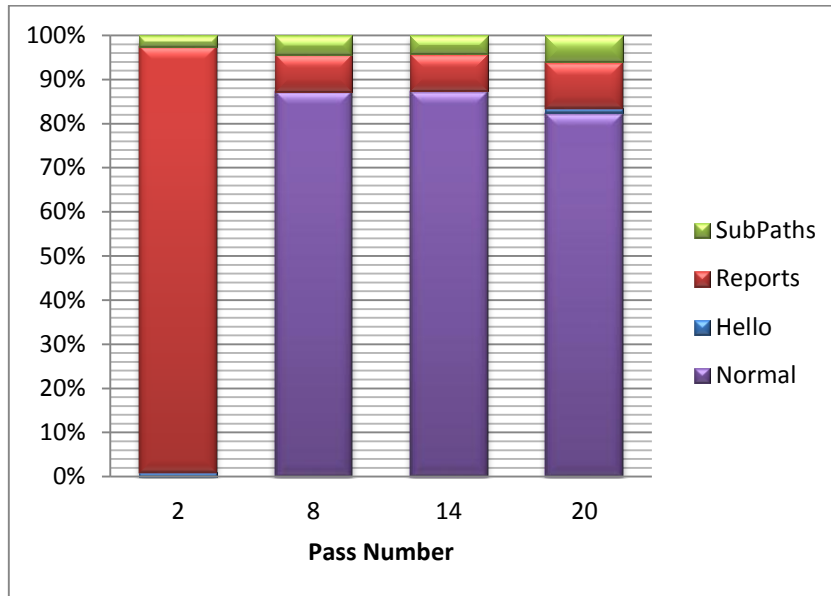


Fig. 6.1 – Communication Overhead without Authentication

Figure 6.2 shows the communicated bytes overhead in comparison to the normal data. It is clear how the overhead imposed by CENTERA starts with a high spike and then continues at a low rate compared to the normal exchanged traffic needing just over 14 passes, or two Report Accumulation Epochs, to be overtaken by them.

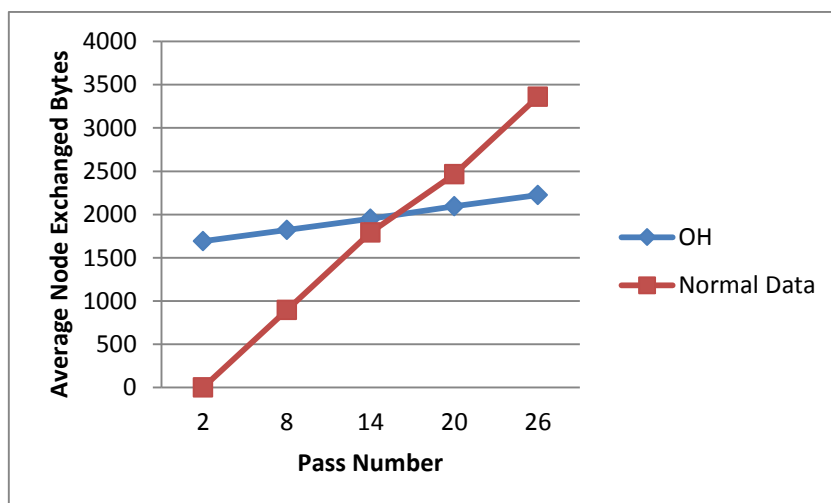


Fig. 6.2 – Overhead with respect to Normal Data

Table 6.2 – Transmission Energy in (μJ) without Authentication

Pass		Hello		Reports		SUB		OH		Normal	
		Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx
2	Network	3,494.40	3,216.00	207,782.40	467,531.36	9,772.80	7,900.64	221,049.60	478,648.00	0.00	0.00
	Av. Per Node	43.68	40.20	2,597.28	5,844.14	122.16	98.76	2,763.12	5,983.10	0.00	0.00
	Worst Case (42)	48.00	53.60	2,822.40	6,619.60	556.80	375.20	3,427.20	7,048.40	0.00	0.00
8	Network	0.00	0.00	18,585.60	15,994.24	9,926.40	8,072.16	28,512.00	24,066.40	193,536.00	168,089.60
	Av. Per Node	0.00	0.00	232.32	199.93	124.08	100.90	356.40	300.83	2,419.20	2,101.12
	Worst Case (32)	0.00	0.00	1,267.20	1,350.72	624.00	423.44	1,891.20	1,774.16	12,902.40	13,807.36
14	Network	0.00	0.00	18,585.60	15,994.24	9,705.60	7,809.52	28,291.20	23,803.76	193,536.00	168,089.60
	Av. Per Node	0.00	0.00	232.32	199.93	121.32	97.62	353.64	297.55	2,419.20	2,101.12
	Worst Case (50)	0.00	0.00	1,910.40	2,068.96	336.00	230.48	2,246.40	2,299.44	19,353.60	21,011.20
20	Network	768.00	3,044.48	18,585.60	15,994.24	10,771.20	8,961.92	30,124.80	28,000.64	145,152.00	126,067.20
	Av. Per Node	9.60	38.06	232.32	199.93	134.64	112.02	376.56	350.01	1,814.40	1,575.84
	Worst Case (32)	9.60	42.88	1,382.40	1,479.36	710.40	493.12	2,102.40	2,015.36	10,483.20	11,256.00

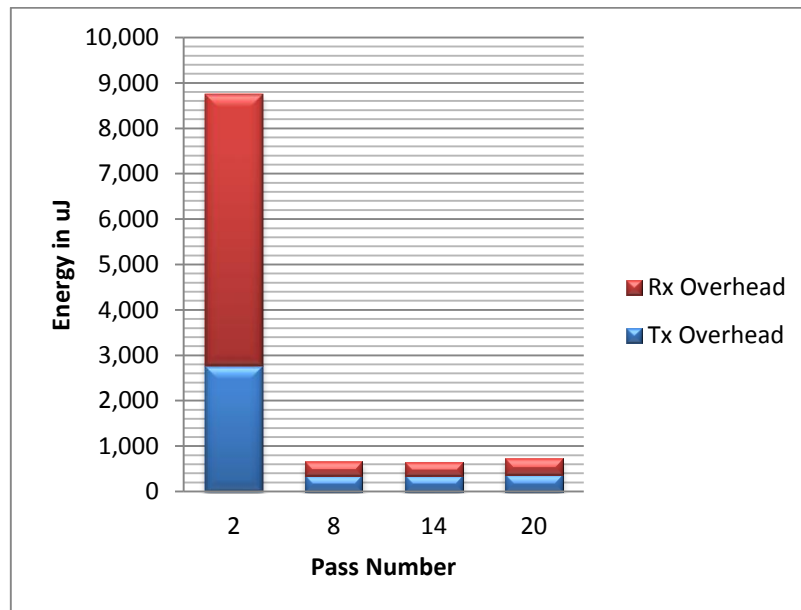


Fig. 6.3 – Energy Overhead without authentication in uJ

Using the energy estimations found in [52], where the energy consumed in MICAz for transmission is $0.6 \mu\text{J}/\text{bit}$ and for receipt is $0.67 \mu\text{J}/\text{bit}$, Table 6.2 contains the exact energy consumed per epoch per period. Figure 6.3 shows the proportion of the

overhead incurred by CENTERA between the transmission energy to that of the reception energy. Two things can be noticed: (1) the reception energy is slightly higher than transmission energy and (2) the energy consumption spikes at the first period to around 8 mJ and then averages to around 650 μ J/period for the remaining network lifetime. Table 6.2 shows that the energy dissipated to transmit and receive normal sensor packets is around 4000 μ J to 4500 μ J/period. Thus, the communication energy overhead is minor when compared to the normal functioning of the sensor network.

Note that, increasing the activity period from 6 to 50, for instance, decreases the overhead even further, as the number of normal packets sent per period is multiplied by approximately 8 and thus the overhead ratio sinks from $650/4500 \approx 14\%$ to around 1.7% overhead.

6.2. Communication Overhead with Authentication

As for the system with a proper authentication scheme incorporated, the overhead imposed by CENTERA will be higher due to the extended messages communicated and the cryptographic technique used. We include the Identity Based—PBC, due to its lightweight processing, short signature (160 bits) and most importantly zero energy and storage to communicate and store keys of every node. This is a direct advantage gained from using identity-based encryption.

Tables 6.3 and 6.4 repeat the analysis of Tables 6.1 and 6.2 in showing the overhead of CENTERA when PBC is incorporated. Similar to the previous results, the overhead is still low as compared to the normal packets communicated by the sensors, as shown in Figure 6.4.

Table 6.3 – Bytes Transmitted and Received with PBC Authentication

Pass		Hello	Reports	SUB	Total OH	Normal	TOTAL
2	Network	14,608.00	476,434.00	8,030.00	499,072.00	0.00	499,072.00
	Av. Per Node	182.60	5,955.43	100.38	6,238.40	0.00	6,238.40
	Worst Case (42)	220.00	6,663.00	426.00	7,309.00	0.00	7,309.00
8	Network	0.00	19,656.00	8,274.00	27,930.00	122,880.00	150,810.00
	Av. Per Node	0.00	245.70	103.43	349.13	1,536.00	1,885.13
	Worst Case (32)	0.00	1,456.00	489.00	1,945.00	9,024.00	10,969.00
14	Network	0.00	19,656.00	7,939.00	27,595.00	122,880.00	150,475.00
	Av. Per Node	0.00	245.70	99.24	344.94	1,536.00	1,880.94
	Worst Case (50)	0.00	2,204.00	273.00	2,477.00	13,632.00	16,109.00
20	Network	8,008.00	19,656.00	9,476.00	37,140.00	92,160.00	129,300.00
	Av. Per Node	100.10	245.70	118.45	464.25	1,152.00	1,616.25
	Worst Case (32)	110.00	1,584.00	600.00	2,294.00	7,344.00	9,638.00

Table 6.4 – Transmission Energy in (μJ) with PBC Authentication

Pass		Hello		Reports		SUB		OH		Normal	
		Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx	Tx	Rx
2	Network	38,438.40	35,376.00	758,342.40	1,706,870.56	19,564.80	21,193.44	816,345.60	1,763,440	0.00	0.00
	Av. Per Node	480.48	442.20	9,479.28	21,335.88	244.56	264.92	10,204.32	22,043.00	0.00	0.00
	Worst Case (42)	528.00	589.60	10,310.40	24,200.40	1,132.80	1,018.40	11,971.20	25,808.40	0.00	0.00
8	Network	0.00	0.00	53,145.60	46,010.24	20,102.40	21,900.96	73,248	67,911.20	331,776.	288,153.60
	Av. Per Node	0.00	0.00	664.32	575.13	251.28	273.76	915.60	848.89	4,147.20	3,601.92
	Worst Case (32)	0.00	0.00	3,571.20	3,816.32	1,296.00	1,173.84	4,867.20	4,990.16	22,118.40	23,669.76
14	Network	0.00	0.00	53,145.60	46,010.24	19,401.60	20,887.92	72,547.20	66,898.16	331,776	288,153.60
	Av. Per Node	0.00	0.00	664.32	575.13	242.52	261.10	906.84	836.23	4,147.20	3,601.92
	Worst Case (50)	0.00	0.00	5,366.40	5,820.96	720.00	659.28	6,086.40	6,480.24	33,177.60	36,019.20
20	Network	8,448.00	33,489.28	53,145.60	46,010.24	22,771.20	25,363.52	84,364.80	104,863.04	248,832	216,115.20
	Av. Per Node	105.60	418.62	664.32	575.13	284.64	317.04	1,054.56	1,310.79	3,110.40	2,701.44
	Worst Case (32)	105.60	471.68	3,878.40	4,159.36	1,574.40	1,457.92	5,558.40	6,088.96	17,971.20	19,296.00

The overhead now remains at around 18% and rises to 28% in periods where keep alive messages are communicated. This increase is partly because of the additional transmissions and receptions incurred on the network, but majorly due to the fact that in

such activity periods, there will be one less period of sending normal packets. Note that, similar to previous analysis, the overhead decreases drastically by increasing the activity period from 6 to 50, from 18% to around 2% and from 28% to around 2.8%.

This stresses the importance of correctly setting the different parameters, and specifically the activity period, which specifies the speed of updating the network and detecting errors at the expense of spending more energy.

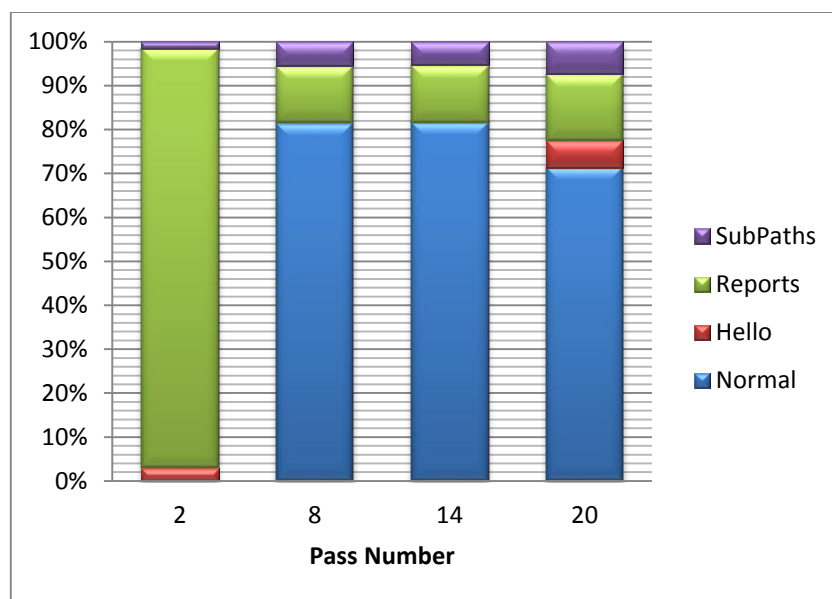


Fig. 6.4 – Communication Overhead with Authentication

The average communicated bytes overhead is compared to the normal exchanged data in Figure 6.5. Similar to Figure 6.2, the overhead imposed by CENTERA in this case also starts with a high spike and then continues at a low rate compared to the normal exchanged traffic. However, in this case the initial spike is much higher needing around 38 passes, or seven Report Accumulation Epochs, to be overtaken by the normal traffic.

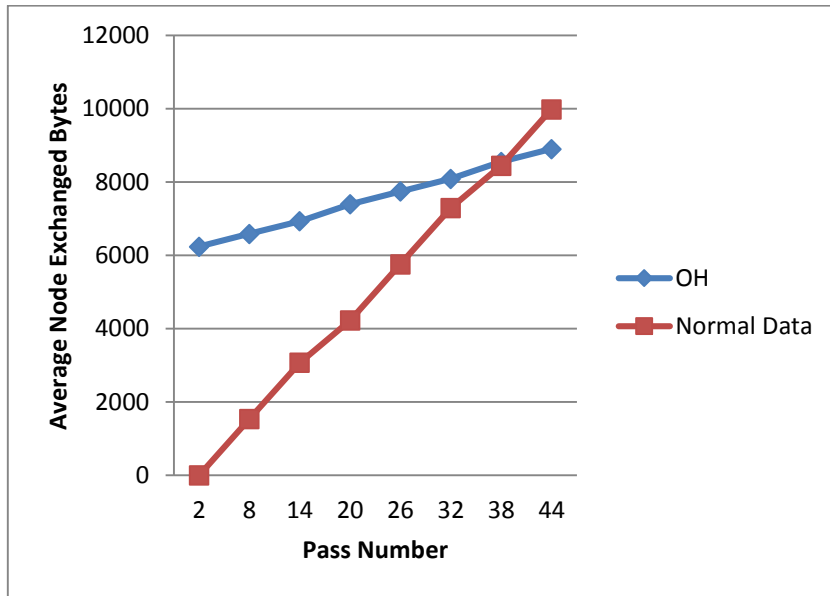


Fig. 6.5 – Overhead with respect to Normal Data

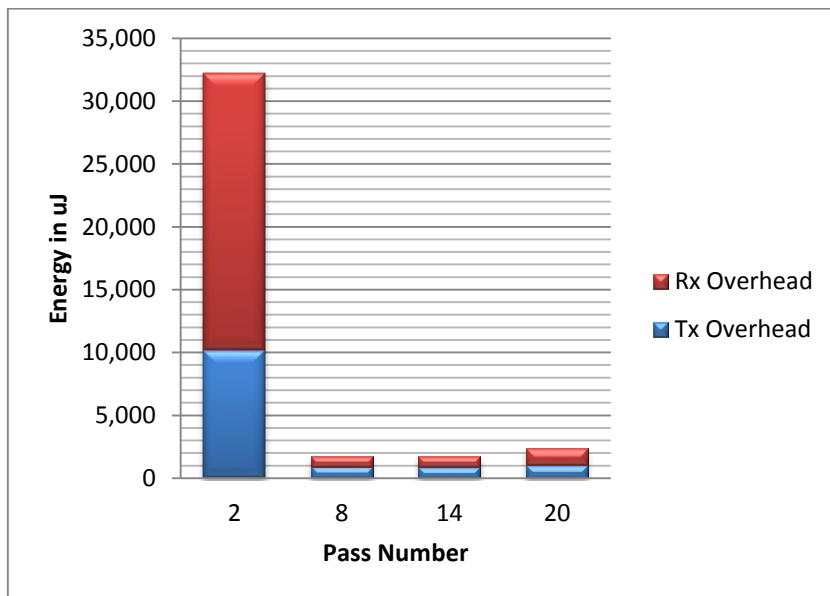


Fig. 6.6 – Energy Overhead with PBC authentication in uJ

Also Figure 6.6 shows similar results to that of Figure 6.3, with the difference that here the energy consumption spikes at the first period to around 30 mJ and then averages to around 1.7 to 2 mJ/period for the remaining network lifetime. This increase is due to the added authentication bytes communicated by the nodes. Table 6.4 shows

that the energy dissipated to transmit and receive normal sensor packets is around 6 mJ to 8 mJ/period. Thus, the communication energy overhead is still minor when compared to the normal functioning of the sensor network.

6.3. Overhead with respect to Activity Period

In order to study the effect of the activity period on the overhead imposed by CENTERA, we repeated our simulations varying the activity period from 5 to 50 passes.

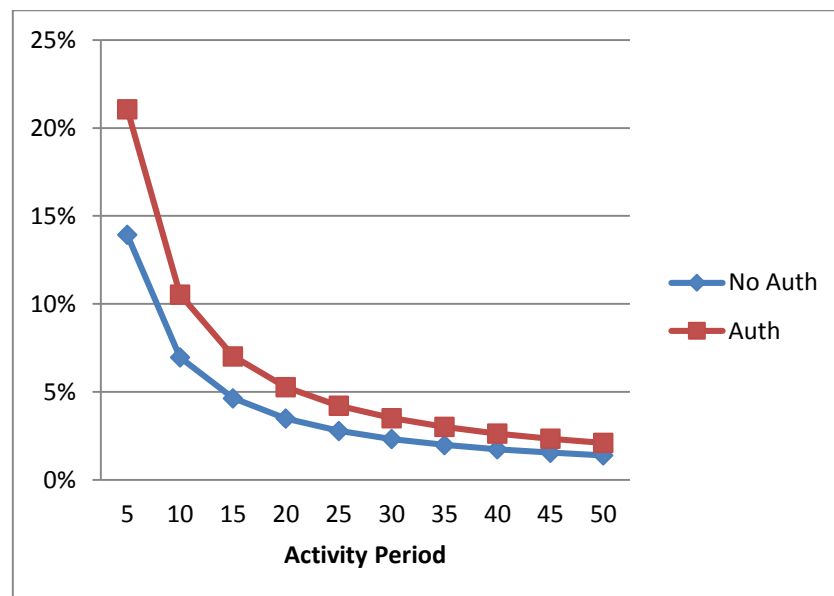


Fig. 6.7 – Average Overhead with respect to the Activity Period

Figure 6.7 shows that the overhead decreases exponentially with the increase of the activity period. For CENTERA with authentication, the overhead decreases from around 22% when the activity period equals 5 to around 2% for activity period of 50. Similarly, the overhead of CENTERA without authentication decreases from around 15% to around 1% as the activity period increases from 5 to 50. One more thing to be

noticed from the figure is the exponential decrease of the overhead imposed by CENTERA as the activity period increases.

This again stresses the significance of setting the different parameters as per the network requirements and the application needs. The choice of the activity period sets the network update and error detection speed at the expense of energy overhead.

6.4. Authentication Overhead

In Table 6.5, we show the effect of the cryptographic techniques to perform the required authentication to secure the WSN in the most hostile environments. We divided the study as per the number of bytes signed and verified by each node to determine the initial sender of the packet; also as per the number of bytes of sources' signatures encrypted and decrypted by subsequent nodes in order to verify the direct hop-by-hop forwarder of the packet.

Table 6.5 – Total Number of Authenticated Bytes

Pass		SIGN	VERIFY	ENC	DEC	TOTAL
2	Network	1,340.00	132,280.00	0.00	2,040.00	135,660.00
	Av. Per Node	16.75	1,653.50	0.00	25.50	1,695.75
	Worst Case (42)	2,029.00	2,029.00	7,309.00	7,309.00	18,676.00
8	Network	9,848.00	37,956.00	22,400.00	18,440.00	88,644.00
	Av. Per Node	123.10	474.45	280.00	230.50	1,108.05
	Worst Case (32)	5,989.00	725.00	10,969.00	1,945.00	19,628.00
14	Network	9,848.00	37,760.00	22,400.00	18,340.00	88,348.00
	Av. Per Node	123.10	472.00	280.00	229.25	1,104.35
	Worst Case (50)	8,849.00	897.00	16,109.00	2,477.00	28,332.00
20	Network	7,768.00	31,356.00	16,800.00	14,740.00	70,664.00
	Av. Per Node	97.10	391.95	210.00	184.25	883.30
	Worst Case (32)	5,098.00	814.00	9,638.00	2,294.00	17,844.00

Note that in the initial phase the major overhead is due to the verification of the

broadcasted neighbor reports, which loaded the network by 132,288 bytes to verify, averaging 1653.5 verified bytes per node. The total number of bytes processed by cryptographic functions reached 135,660 bytes in the network averaging 1695.75 bytes per node in the initial phase. In subsequent phases, this number dropped to around 88,000 bytes in the whole network, averaging to around 1100 bytes per node. Note that the last period, where there is one less period to send normal packets, the total is just 70,664 bytes to authenticate, which clearly shows that the majority of the processing overhead is spent authenticating the normal sensor messages.

One thing that can be noticed is that the worst case in the normal phases is much higher than that of the initial phase. This is clearly described by the absence of normal packets in the initial phase. This gives an indication of the huge load that the closer nodes to the BS have to endure due to the nature of such networks.

The majority of the cryptographic overhead is due to the normal packets communicated and not due to the epochs of CENTERA. This conforms to the previously gathered results where we noticed that the additional number of packets sent by CENTERA is in the range of 12% to 17% for the chosen activity period.

6.5. Authentication Advantage – A Broadcasting Node

To further show the advantage of authentication in a trust based system for energy efficiency, we introduced into our system a broadcasting attacker node while changing its identity and calculated the cumulative number of bytes exchanged until each pass.

In this scenario, node 11 broadcasts normal packets to its neighbors at a high rate while changing its identity, in order that its packets are forwarded by its neighbors into the system. As explained previously, CENTERA forces nodes to only forward the

packets of their designated UL neighbors.

Table 6.6 – Cumulative Number of Bytes Up to each Pass

Cumulative Number of Bytes Up to each Pass						
Pass		2	8	14	20	
Broadcasting Rate without authentication	10	Network	134,630.00	367,907.00	601,184.00	834,461.00
		Av. Per Node	1,682.88	4,598.84	7,514.80	10,430.76
		Worst Case	1,973.00	12,719.00	23,465.00	34,211.00
	50	Network	136,404.00	465,790.00	795,176.00	1,124,562.00
		Av. Per Node	1,705.05	5,822.38	9,939.70	14,057.03
		Worst Case	2,075.00	21,463.00	40,851.00	60,239.00
	100	Network	135,368.00	569,250.00	1,003,132.00	1,437,014.00
		Av. Per Node	1,692.10	55,927.35	110,162.60	164,397.85
		Worst Case	1,997.00	34,356.00	66,715.00	99,074.00
Normal Case w/ auth	Network	499,072.00	735,793.00	886,268.00	1,015,568.00	
	Av. Per Node	6,238.40	9,197.41	11,078.35	12,694.60	
	Worst Case	7,309.00	25,543.00	41,652.00	51,290.00	

We repeat this scenario while changing the broadcasting rate of node 11, from ten times the normal rate of normal packets, to 50 and 100 times. Table 6.6 shows the cumulative number of bytes exchanged for the whole network of 81 nodes, the average number of bytes per node, and the worst case, in each broadcasting rate and compare them to the normal case of CENTERA with authentication. It is clear from the table how the overhead of the authentication is offset by the broadcasting attacker in as low as three and four passes of the system for the 100 and 50 times broadcasting rate cases. Also in the low rate of ten times broadcasting rate, the overhead is closing up in four passes in the table. Figure 6.8 displays the network exchanged bytes and it is clear how the authentication overhead is overcome very quickly by one attacker.

Note that in CENTERA this attack will be limited in time and space. As a direct benefit of authentication, node 11 is unable to deceive its neighbors by using their UL node ID and thus those neighbors drop the packets of the attacker; the attack is thus

limited in space. In the next pass, as the packets are authenticated, the BS detects the high sending rate of node 11, and isolates it to end its negative effect from the system; the attack is thus limited in time.

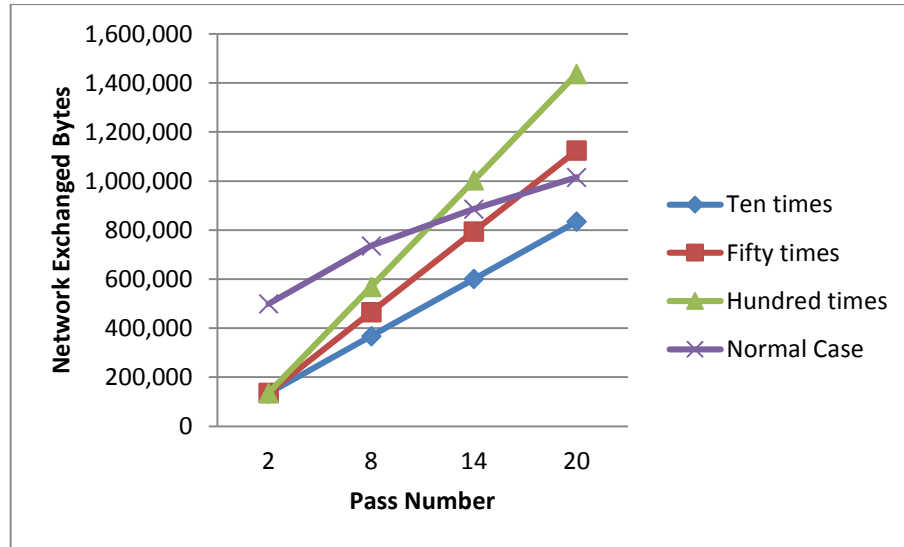


Figure 6.8 – Cumulative Number of Exchange Bytes in the Network up to each Pass

6.5. Network Lifetime Calculations

Finally, we calculate the network life-time in terms of the first node dead (FND) and the residual energy after multiple rounds of data exchanged. We consider the energy to compute one byte during PBC authentication to be $w\%$ of the energy to receive one byte and vary w between 10%, 50%, 100%, and even 1000%. Note that it is well known that the energy to compute is much lower than the energy to transmit/receive a byte, thus our chosen values of w are all high assumptions to check the worst case for the energy overhead; especially the last assumption where we consider the computation energy on one byte to be 10 times that to transmit/receive one.

We consider that each sensor node has a small battery of 15,000 J and perform the life-time analysis as follows. Table 6.7 shows the network life-time calculations

while varying the value of w from 10% to 1000%. The results are the average energy consumption in the first pass and the average in the following passes in mJ. Also it shows that the first node dies after over 203,073 passes when $w = 10%$ and over 83,482 passes even when considering the energy to compute one byte to be equal to that to receive one byte. As for the extreme case where the energy to compute one byte is assumed to be ten times that to receive one byte, the first node dies after over 12,117 passes.

Table 6.7 – Network Life-time Calculations

W	10%		50%		100%		1000%	
	First Pass	Av	First Pass	Av	First Pass	Av	First Pass	Av
Energy Node Av. (mJ)	33.16	9.61	36.79	11.83	41.34	14.59	123.14	64.37
Energy Worst Case (mJ)	47.79	73.86	87.83	120.89	137.88	179.68	1,038.81	1,237.81
Av. Node Dead (Passes)	1,560,206.07		1,268,336.34		1,027,959.17		233,025.89	
FND (Passes)	203,073.02		124,076.20		83,482.07		12,117.34	

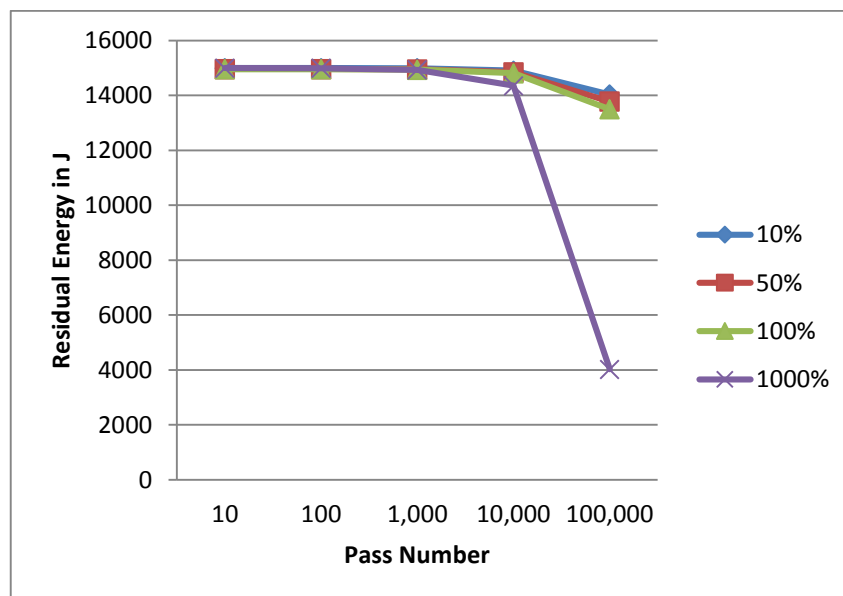


Fig. 6.9 – Average Node Residual Energy in J

Figure 6.9 shows the average residual energy in every node starting with a full 15,000 J battery at powers of ten passes.

The figure shows the low energy consumption of CENTERA, since even when considering that the energy to compute one byte equals that to receive a byte, the average residual energy after 100,000 passes is still 13,497 J which is around 90% of the battery charge; and the battery is still over 93.5% when $w = 10\%$. As for the extreme case when $w = 1000\%$, the average residual energy in a node is a bit over 4000 J that is around 27%.

Figure 6.10, shows the residual energy for the most active node, which is most logically one of the closest to the BS. It can be seen that even in the extreme case of $w = 1000\%$, the first dead node is seen at over 12,000 passes, and over 80,000 passes for $w = 1000\%$.

Also, the figure shows that assuming $w = 10\%$, the residual energy in the worst case is still over 50% after 100,000 passes, and around 17% for $w = 50\%$.

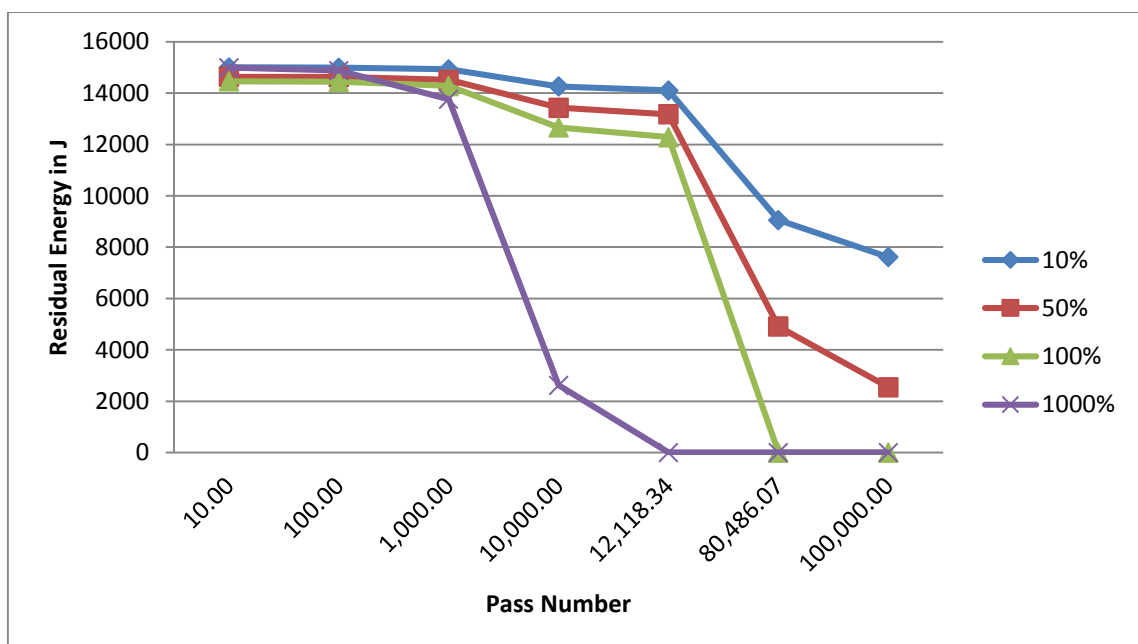


Fig. 6.10 – Worst Case Residual Energy in J

CHAPTER 7

CONCLUSION AND FUTURE DIRECTIONS

In this dissertation, we presented CENTERA, a CENTralized Trust-based Efficient Routing protocol with an appropriate Authentication scheme for wireless sensor networks (WSN), which periodically sends readings and sensed data to a powerful sink BS. We showed how CENTERA provides secure routing and a trusted network where the bad nodes are isolated and their attacks eliminated. We classified different types of bad nodes, some of which are malicious, incompetent, non-cooperative selfish, broadcasting, outsider, and impersonating nodes that affect the routing functionality of the network.

CENTERA utilizes the centralized approach, where the more powerful BS periodically accumulates simple counter observations from the sensor nodes and decides on the network topology and routes, and isolates the bad nodes. The BS calculates several quality metrics and two trust levels of each node and uses an effective banning system to isolate the different bad nodes from the network. Also, CENTERA uses a very efficient method to distribute the routing information to every node. The nodes forwards only through their next hop DL neighbor and forward the packets of their UL neighbors, only as indicated by the BS and drop any other packet.

For the proper functioning of our routing protocol and the necessary validation of the nodes to each other and to the base station, CENTERA uses a secure and efficient authentication scheme suitable for the extremely limited sensor nodes in WSNs providing acceptable security levels while requiring minimal processing power and data transmission overhead. For this essential issue, we discussed the different authentication

techniques suitable for the severely constrained sensor nodes in WSNs, and addressed three main categories based on symmetric cryptography, asymmetric cryptography, and hybrid techniques using both cryptographic methods. We discussed each category and deduced that RC5 is among the most appropriate to use in symmetric key techniques and PBC or IBE-ECC are so far the most promising asymmetric key cipher. We investigated the different factors affecting the choice of the authentication technique to be used depending on the energy and memory requirements of the cipher as well as the cipher security strength, with slightly more stress on the former. Also the choice of which category to use depends on the environment of the network, the attacker model, and the sensitivity of the data being sensed.

We implemented CENTERA using TinyOS and proved its correctness in providing secure routing information through trusted paths. Also, in CENTERA, some nodes were put on probation to observe closely while bad nodes were isolated for a specific time depending on their history, and then given another chance to try to improve.

CENTERA was proven to be a scalable trust based and balanced routing protocol protecting the network and sensor nodes from most known attacks while imposing minimal overhead levels, depending on the different parameters and assumptions. Depending on the authentication technique used and the energy required to process the cryptographic techniques, CENTERA's energy calculations for the first node dead (FND) and the remaining residual energy proved to provide the WSN with a long network lifetime.

Future work would focus majorly on several research directions including:

- 1- Further enhancing and adding more intelligence to the BS analysis algorithm to more efficiently perform all the complex tasks it is entitled to

do.

- 2- Checking the inclusion of secure positioning or other positioning system to better validate nodes' neighborhood and detect colluding nodes.
- 3- Extending the model to include hierarchical topologies including cluster-heads and to allow for node to node interactions.
- 4- Accounting for nodes mobility. CENTERA can account for mobility with minor adjustments including repeating the Neighbor Discovery Epoch periodically or at every change of the neighborhood. Many scenarios should be simulated and parameters studied in order to choose the most efficient process to adopt.
- 5- Performing an extensive study on the hybrid technique to come up with the parameters affecting the energy and memory overhead and assessing the general functionality/applicability and security of such a technique.
- 6- Extending this work to the less energy constrained devices in mobile ad hoc networks (MANETs). In MANETs, there is the possibility to use a hybrid version of the routing protocol utilizing the centralized approach while demanding extra work from the devices. Introductory work on this topic was published in [53].

REFERENCES

- [1] Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. A Survey on Sensor Networks. *IEEE Commun. Mag.* **2002**, *40*, 102–114.
- [2] Karlof, C.; Wagner, D. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad Hoc Netw.* **2003**, *1*, 293–315.
- [3] Srinivasan, A.; Teitelbaum, J.; Wu, J.; Cardei, M.; Liang, H. Reputation and Trust-based Systems for Ad Hoc and Sensor Networks. In Algorithms and Protocols for Wireless, Mobile, Ad Hoc Networks; *Azzedine Boukerche*; Wiley-IEEE Press: 2009; pp. 375-404.
- [4] Tanachaiwiwat, S.; Dave1, P.; Bhindwale, R.; Helmy, A. Location-centric Isolation of Misbehavior and Trust Routing in Energy-constrained Sensor Networks. In Proceedings of the 2004 IEEE International Conference on Performance, Computing, and Communications, Phoenix, Arizona, USA, April 15-17, 2004.
- [5] Huang, L.; Li, L.; Tan, Q. Behavior-Based Trust in Wireless Sensor Network. *Adv. Web Netw. Technol. Appl. LNCS* **2006**, *3842*, 214–223.
- [6] Tajeddine, A.; Chehab, A.; Kayssi, A. CENTER: A Centralized Trust-Based Efficient Routing Protocol for Wireless Sensor Networks. In Proceedings of the Tenth Annual Conference on Privacy, Security and Trust, Paris, France, 16–18 July 2012.
- [7] Al-Karaki, J.; Kamal, A. Routing Techniques in Wireless Sensor Networks: A Survey. *IEEE Wireless Communications* **2004**, doi:10.1109/MWC.2004.1368893.
- [8] Dikondwar, D.; Krishna, R. Survey: Energy-Efficient and Trust-Aware Routing Techniques for WSN. *International Journal of Engineering Research & Technology (IJERT)*. **2013**, *2*, 1-6.

- [9] Raghunandan, G.; Lakshmi, B. A Comparative Analysis of Routing Techniques for Wireless Sensor Networks. In Proceedings of the National Conference on Innovations in Emerging Technology, India, 17-18 February, 2011, pp. 17-22.
- [10] Yu, Y.; Li, K.; Zhou, W.; Li, P. Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures. *J. Netw. Comput. Appl.* **2012**, *35*, 867–880.
- [11] Zahariadis, T.; Leligou, H.; Karkazis, P.; Trakadas, P. Energy efficiency and implementation cost of trust aware routing solutions in WSNs. In Proceedings of the 14th Panhellenic Conference on Informatics 2010, IEEE Computer Society, pp 194-198.
- [12] Hoffman, K.; Zage, D.; Rotaru, C. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys* 2009; *42*(1):1–31.
- [13] Lopez, J.; Roman, R.; Alcaraz, C. Analysis of security threats, requirements, technologies and standards in wireless sensor networks. *Foundations of Security Analysis and Design V* 2009; *5705*:289–338.
- [14] Xiao, B.; Yu, B. Detecting selective forwarding attacks in wireless sensor networks. In Proceedings of the 20th international parallel and distributed processing symposium (IPDPS); 2006. pp. 1–8.
- [15] Xiao, B.; Yu, B.; Gao, C. Chemas: identify suspect nodes in selective forwarding attacks. *Journal of Parallel and Distributed Computing* 2007; *67*(11):1218–30.
- [16] Khalid, O.; Khan, S.; Madani, S.; Hayat, K.; Khan, M.; Min-Allah, N.; Kolodziej, J.; Wang, L.; Zeadally, S.; Chen, D. Comparative study of trust and reputation systems for wireless sensor networks. *Security and Communication Networks* (2012), *6*(6), 669-688. Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.597.

- [17] Menezes, A.; Oorschot, V.; Vanstone, S. Handbook of Applied Cryptography. CRC Press, October 1996 – 5th reprinting, Aug. 2001.
- [18] Schurgers, C.; Srivastava, M. Energy Efficient Routing in Wireless Sensor Network. In Proceedings of the MILCOM'01, Vienna, VA, USA, 28–31 October 2001; pp. 357–361.
- [19] Momani, M.; Challa, S.; Aboura, K. Modelling Trust in Wireless Sensor Networks from the Sensor Reliability Prospective. *Innov. Algorithms Tech. Ind. Electron. Telecommun.* **2007**, 317–321, doi: 10.1007/978-1-4020-6266-7_57.
- [20] Yadav, K.; Srinivasan, A. iTrust: An Integrated Trust Framework for Wireless Sensor Networks. In Proceedings of the ACM Symposium on Applied Computing, Sierre, Switzerland, 22–26 March 2010; pp. 1466–1471.
- [21] Momani, M. Trust Models in Wireless Sensor Networks: A Survey. *Recent Trends Netw. Secur. Appl.* **2010**, 89, 37–46.
- [22] Chen, X.; Makki, K.; Yen, K.; Pissinou, N. Sensor Network Security: A Survey. *IEEE Commun. Surveys Tutor.* **2009**, 11, 52–73.
- [23] Yu, Y.; Li, K.; Zhou, W.; Li, P. Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures. *J. Netw. Comput. Appl.* **2012**, 35, 867–880.
- [24] Muruganathan, S.; Ma, D.; Bhasin, R.; Fapojuwo, A. A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks. *IEEE Radio Commun.* **2005**, doi:10.1109/MCOM.2005.1404592.
- [25] Akkaya, K.; Younis, M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Netw.* **2005**, 3, 325–349.
- [26] Manjeshwar, A.; Agrawal, D.P. TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. In Proceedings of the International Parallel and

- Distributed Processing Symposium, San Francisco, CA, USA, 23–27 April 2000; Volume 3.
- [27] Manjeshwar, A.; Agrawal, D.P. APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In Proceedings of the International Parallel and Distributed Processing Symposium, Ft. Lauderdale, FL, USA, 15–19 April 2001; Volume 2.
- [28] Chang, J.H.; Tassiulas, L. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Trans. Netw. (TON)* **2004**, 609–619, doi:10.1109/TNET.2004.833122.
- [29] Yu, Y.; Govindan, R.; Estrin, D. *Geographical and Energy Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks*; Technical Report ucla/csd-tr-01-0023; UCLA Computer Science Department: Los Angeles, USA, 2001.
- [30] Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: A survey. *Comput. Netw.* **2002**, 38, 393–422.
- [31] Ganesan, D.; Govindan, R.; Shenker, S.; Estrin, D. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2001**, 5, 11–25.
- [32] ONF Market Education Committee. Software-defined networking: The new norm for networks. ONF White Paper (2012). Available online: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf> (accessed on 16 January 2015).
- [33] Tajeddine, A.; Kayssi, A.; Chehab, A.; Elhadj, I. Authentication Schemes for Wireless Sensor Networks. In Proceedings of the 17th IEEE Mediterranean Electrotechnical Conference (MELECON), Beirut, Lebanon, 13–16 April 2014.

- [34] Rehman, S.; Bilal, M.; Ahmad, B.; Yahya, K.; Ullah, A.; Rehman, O. Comparison Based Analysis of Different Cryptographic and Encryption Techniques Using Message Authentication Code (MAC) in Wireless Sensor Networks (WSN). *International Journal of Computer Science Issues*, **2012**, vol. 9, no. 1.
- [35] Si, L.; Ji, Z.; Wang, Z. The application of symmetric key cryptographic algorithms in wireless sensor networks. *Physics Procedia*, vol. 25, pp. 552–559, Jan 2012.
- [36] Söderlund, R. Energy efficient authentication in wireless sensor networks. M.S. Thesis, Department of Computer and Information Science, Linköping University, Linköping, Sweden, 2006.
- [37] Noroozi, E.; Kadivar, J.; Shafiee, S. Energy Analysis for Wireless Sensor Networks. In *Proc. 2nd International Conference on Mechanical and Electronics Engineering (ICMEE 2010)*, 2010.
- [38] Mani, D.; Nishamol, P. A Comparison Between RSA And ECC In Wireless Sensor Networks. *International Journal of Engineering Research & Technology*, Vol. 2, Issue 3, March 2013.
- [39] Yasmin, R.; Ritter, E.; Wang, G. An Authentication Framework for Wireless Sensor Networks Using Identity-Based Signatures: Implementation and Evaluation. *IEICE Transactions on Information and Systems*, E95-D (1), pp. 126-133, 2012.
- [40] Yussoff, Y.; Hashim, H.; Baba, M. Analysis of Trusted Identity Based Encryption (IBE-Trust) Protocol for Wireless Sensor Networks. *IEEE Control and System Graduate Research Colloquium (ICSGRC 2012)*.
- [41] Saab, S.; Kayssi, A.; Chehab, A. A Decentralized Energy-Aware Key Management Scheme for Wireless Sensor Networks. In *Proc. 6th International Conference for Internet Technology and Secured Transactions, (ICITST-2011)*, December 11-14, 2011, Abu Dhabi, UAE.

- [42] Ganesan, P.; Venugopalan, R.; Peddabachagari, P.; Dean, A.; Mueller, F.; Sichitiu, M. Analyzing and Modeling Encryption Overhead for Sensor Network Nodes. In Proceeding of the 1st ACM international workshop on Wireless sensor networks and application, San Diego, California, USA, September 2003.
- [43] Rivest, R. The RC5 Encryption Algorithm. In *Fast Software Encryption*, pp. 86-96. Springer Berlin Heidelberg, 1995.
- [44] Lee, J.; Kapitanova, K.; Son, S. The Price of Security in Wireless Sensor Networks. *Computer Networks* 54, no. 17 (2010): 2967-2978.
- [45] Gaubatz, G.; Kaps, J.; Sunar, B. Public Key Cryptography in Sensor Networks- Revisited. In 1st European Workshop on Security in Ad-Hoc and Sensor Networks, ESAS 2004.
- [46] Koblitz, N. Elliptic Curve Cryptography. *Mathematics of Computation* 48, no. 177, 1987.
- [47] Amin, F.; Jahangir, A.; Rasifard, H. Analysis of Public-Key Cryptography for Wireless Sensor Networks. in *Proc. Of World Academy of Science, Engineering and Technology*, 2008.
- [48] J. Zheng, J. Li, M. Lee, and M. Anshel, "A Lightweight Encryption and Authentication Scheme for Wireless Sensor Networks", *Int. J. Security and Networks*, Vol. 1, No 3/4, 2006.
- [49] TinyOS. Available online: <http://www.tinyos.net/> (accessed on 16 January 2015).
- [50] TinyPairing: A Pairing-Based Cryptographic Library for Wireless Sensor Networks. Available online: <http://www.cs.cityu.edu.hk/~ecc/TinyPairing/> (accessed on 16 January 2015).
- [51] Xiong, X.; Wong, D.S.; Deng, X. TinyPairing: A Fast and Lightweight Pairing-based Cryptographic Library for Wireless Sensor Networks. In Proceedings of the

- IEEE Wireless Communications & Networking Conference (IEEE WCNC10)*, Sydney, Australia, 18–21 April 2010.
- [52] Meulenaer, G.D.; Gosset, F.; Standaert, F.X.; Pereira, O. On the energy cost of communication and cryptography in wireless sensor networks. In Proceedings of the IEEE International Conference on Wireless Mobile Computing, Networking, Communication, Avignon, France, 12–14 October 2008; pp. 580–585.
- [53] Tajeddine, A.; Chehab, A.; Kayssi, A. H-TRACE: A hybrid energy-aware routing scheme for mobile ad hoc networks. In Proceedings of the International Conference on Energy Aware Computing (ICEAC), 2011, pp. 1-6.
- [54] Powell, O.; Moraru, L.; Seigneur, JM. TrustMIX: Trustworthy MIX for Energy Saving in Sensor Networks. arXiv preprint arXiv:0705.2313 (2007).
- [55] El Maliki, T.; Seigneur, JM. Reliability and Survivability of Wireless Sensor Network Using Security Adaptation Reference Monitor (SARM). In Proceedings of the Fifth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE), 2011, pp. 71-76.
- [56] Zhan, G.; Shi, W.; Deng, J. Tarf: A trust-aware routing framework for wireless sensor networks. In *Wireless Sensor Networks*, 2010, Springer Berlin Heidelberg, pp. 65-80.
- [57] Sha, K.; Du, J.; Shi, W. WEAR: a balanced, fault-tolerant, energy-aware routing protocol in WSNs. *International Journal of Sensor Networks* 1, no. 3 (2006): pp. 156-168.