DEMOCRATIC REPUBLIC OF ALGERIA PEOPLE
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
UNIVERSITY MOHAMED BOUDIAF - M'SILA

FACULTY: mathematics and informatics
DEPARTEMENT of computer science
N°: ……

DOMAIN: Mathematics and Informatics
FIELD: computer science
SUB-FIELD: Information Communication Technologies

A Dissertation in Fulfillment
for the Requirement of the Degree of MASTER

By: Loubna BOUSSAG

SUBJECT:

Implementation and Simulation of Security Protocols for Wireless Sensor Networks (WSNs)

Defended publicly on: 07/06/2017, to the jury:

Board of Examiners

Meftah Lakehal	University of M'sila	Chairman
Noureddine Chikouche	University of M'sila	supervisor
Mohamed Benouis	University of M'sila	examiner

Academic year: 2016/2017

Dedication

To my parents, for their endless love, support and encouragement

To my family.

ACKNOWLEDGEMENTS

Foremost, I am grateful to the almighty ALLAH for helping me to complete this research paper.

I would like to express my sincere gratitude to my advisor CHIKOUCHE Noureddine, for all his guidance, and for his timely and highly up to the mark feedback and support.

I would like to thank my loved ones, who have supported me throughout the entire process, both by keeping me harmonious and helping me putting pieces together. I will be grateful forever for your love. My great friends, and all the doctors and professors for teaching us through our college years. I also place on record, my sense of gratitude to all those who, directly or indirectly, have lent their helping hand in this research paper. In addition, to all my family for standing by me in every day of my life. To all of you, my love and respect.

Table of Contents

List of Figures

Chapter 1: WIRELESS SENSOR NETWORKS (WSNs)

Chapter 2: SECURITY FOR WSNs

Chapter 3: SECURITY PROTOCOLS FOR WSNs

Chapter 4: IMPLEMENTATION AND SIMULATION

List of Tables

GENERAL INTRODUCTION

In recent years, the rise of wireless technologies has opened new prospects in the field of telecommunications. Compared to the wired environment, the wireless environment enables users to be flexible and easy to manipulate information through mobile computing units (laptop, PDA, sensor, etc.). wireless networks can be divided into two categories: network with fixed infrastructure pre-existing, and networks without infrastructure (ad-hoc).

With technical advances in terms of performance and miniaturization, realized in the electromechanical microsystems (MEMS: microcontroller, transceiver RF ...) and wireless communications, a new variant of ad-hoc networks has been created in order to offer economically interesting solutions for the remote monitoring and the data processing in complex and distributed environments: Wireless sensor networks(WSN) that are composed of small electronic devices. These elements are autonomous, equipped with sensors and able to communicate with each other wirelessly. In addition, they collaborate with each other to form a wireless sensors network capable of supervising a region or phenomenon of interest, providing useful information by combining the measurements taken by the various sensors and then communicating them via a wireless support to a remote control center.

In the WSN, in which the network must be secured against external and internal attacks, and against resource depletion (depletion of energy resources), these attacks can harm their work and prevent their proper deployment goal. Therefore, security is an important dimension for these networks. In order to assure the security services, it is necessary for the communicating nodes to share cryptographic keys for encryption and authentication. The latter make it possible to keep the information transmitted through the networks secret. Most security approaches currently available fall into one of the following classes: symmetric cryptography approach or asymmetric cryptography. This offers a better resistance against the compromise of nodes and allows the passage to scale but requires in part an overhead on the software and hardware of the nodes. among the works that have targeted this approach and in the context of WSN there is Tiny Elliptic Curve Cryptosystem (TinyECC) [25]. This library provides digital signature schema (ECDSA) [15], the key exchange protocol (ECDH) [3], and Elliptic Curve Integrated Encryption Schema (ECIES) [34]. recently Moon et al proposed a Mutual Entity Authentication Protocol based on ECDSA [4].

In this research paper we propose a study over some different security protocols for WSNs which are: ECDH, ECDSA and protocol of Moon et al. which is our choice for the implementation, our major objective is simulating and comparing these protocols by relying on terms of performance such as energy consumption, size memory and execution time. The protocols performances are evaluated by using the simulator: TOSSIM and Avrora.

Outline

Chapter 1:

Presentation of general concepts related to the field of WSN through identify their characteristics, domains application, conception constraints, and their architecture.

Chapter 2:

The second chapter includes the main concepts of security for WSN such as: objectives, attacks in the WSN, then, we present security mechanisms for this type of network.

Chapter 3:

This chapter includes security protocols description with some different categories

Chapter 4:

The fourth chapter present a comparative study of the security protocols by evaluating their performance based on the simulation under TOSSIM and Avrora using the TinyOS environment. Finally, this work end with a conclusion of the obtained results and the future perspectives.

CHAPTER 1

WIRELESS SENSOR NETWORKS

(WSNs)

1.1 Introduction

Recently The progress made in the field of microelectronics, micromechanics, and wireless communication technologies, have made it possible to produce components with a reasonable cost of a few cubic millimeters of volume. The latter, called micro-sensors which is used to sensor a physical measure and transfer it to a digital measure.

Therefore, the micro-sensors are real embedded systems. They are able to generate and exchange data in an autonomous way and completely transparent way to the users forms a wireless sensor network (WSN). these networks are currently a new domain, involving of development, emerging from innovation of communication technologies.

In this chapter, we will give an overview about WSN by introducing some description of their general concepts; firstly, we will start by the definition of networking, wireless network wireless sensor network, sensor node and its architecture. then, we will specify the WSN objectives, characteristics, domains application, conception constraints.as well as, the WSN communications such as (protocol stack, standards technologies). finally, we will conclude the chapter by giving a small conclusion.

1.2 Networking

A computer network is a set of equipment interconnected in order to exchange information by analogy with a net (a network is a "small net") we call the node endpoint of a connection, which can be an intersection of several connections, (a computer, a router, a hub, a switch). Computer networks are classified according to their scope such as: personal network (PAN), local area network (LAN), metropolitan network (MAN), and wide area network (WAN) [27].

1.3 Wireless networks

A wireless network are computer networks that connects different stations or systems between them by radio waves. It may be associated with a telecommunications network to interconnections between nodes as mentioned in: [27]. The most currently used standard for wireless networks is the standard IEEE802.11, also known as Wi-Fi.

The next figure 1.1 illustrates the different categories of wireless network, the standards used and their area of coverage.

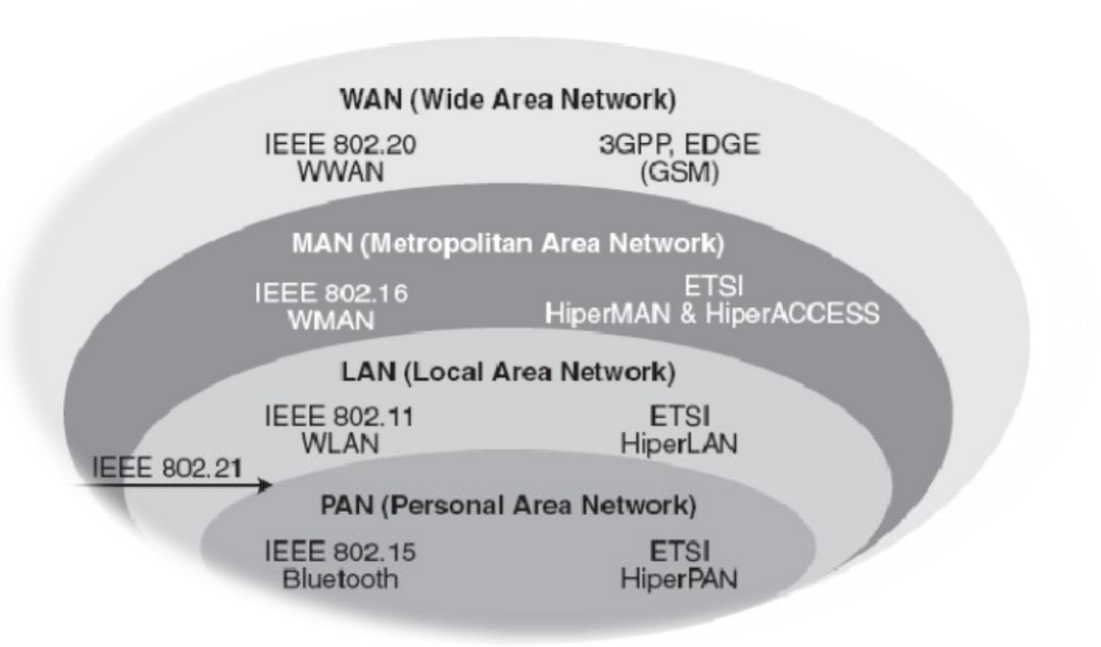

Figure 1.1 Different categories of wireless network [18]

1.4 Wireless sensor network (WSN)

1.4.1 Definition

Wireless Sensor Networks (WSN) is a wireless network Large-scale distributed system with a large number of very small devices, autonomous, commonly called "wireless sensors", or simply "sensors". Each node of these networks is able to monitor its environment and responding, if necessary, by sending the collected information to one or more-point collection, using a wireless connection.

Sensors are very limited, autonomous, extremely small sized devices, capable of processing and transmitting information via radio waves (Wi-Fi or ZigBee 3 for example), to another entity (sensors, treatments ...) over a limited distance of few meters as mentioned in [38]. The nodes have the ability to play the role of routers. A sensor analyzes its environment and propagates the data collected to the sensor belonging to its area of coverage.

1.4.2 Objective of WSN:

The main objectives of wireless sensor networks generally depend on applications. however, the following tasks are common to several applications [19]:

- Determine the values of some parameters according to a given situation. for example, in an environmental network, we can search: the temperature, the pressure, the amount of sunlight, and relative humidity in a number of sites, etc.
- Detect the occurrence of events of interest and estimate the parameters of events detected.

In traffic control network, it may be desired to detect vehicles movement through intersection and estimate its speed and direction.

❖ Classify the detected objects. In a traffic network, whether is a vehicle a car, a bus, etc.

1.5 Sensor node

1.5.1 Definition

A sensor is an electronic device in charge [18] of transforming an environmental physical measure observed in a generally electrical measure that will be its turn translated into a binary data exploitable a comprehensible by a system information and to communicate it to a control center via a base station. This measure can be in different types such as (temperature, humidity, brightness, acceleration, distance, movements, position, pressure of gas, vision (image capture), sound ……etc.).

Recently there are a large type of sensors with various and varied functionalities. The figure 1.2 below shows some examples of sensors

Figure 1.2 Example of sensors [38]

1.5.2 Sensor node characteristics

A sensor node is equipped with large number of characteristics which allows us to differentiate it from the other types of sensors i.e. classify the sensors in categories for example (powerful sensors or low sensors) and associating each sensor with its application domain.

In the following example (the table1.1) we summarize some type of sensors with their characteristics in term of processor, radio, memory, and battery voltage.

Model or type	Microcontroller (processor)	Radio Frequency band	RAM Memory	Flash Memory	Battery voltage
Mica2	ATmega128(8-bit)	868/916 MHz	4KB	128KB	2.7-3.3 V
MicaZ	8MHz ATmega128 (8-bit)	2400 - 2483.5 MHz	4KB	128KB	2.7-3.3 V
TelosB	1MHz TI MSP430 (16-bit)	2400 - 2483.5 MHz	10KB	48KB	1.8-3.6 V
Imote2	Intel PXA271(32-bit)	2400.0 -2483.5 MHz	32MB	32MB	3.2-4.5 V

Table 1.1 Some sensor characteristics [6]

1.5.3 Sensor node architecture

The basic architecture of a sensor node is composed of four main units depending on the field of application. namely as follow:

- The sensors unit (an element that is responsible for measuring the external environment.)
- The processing unit
- The transmission unit (sender / receiver)
- The energy control unit

this architecture may also contain other modules, such as:

- a location search system (GPS)
- an energy generating system (solar cell)
- a mobility system to move the micro-sensor into the capture environment

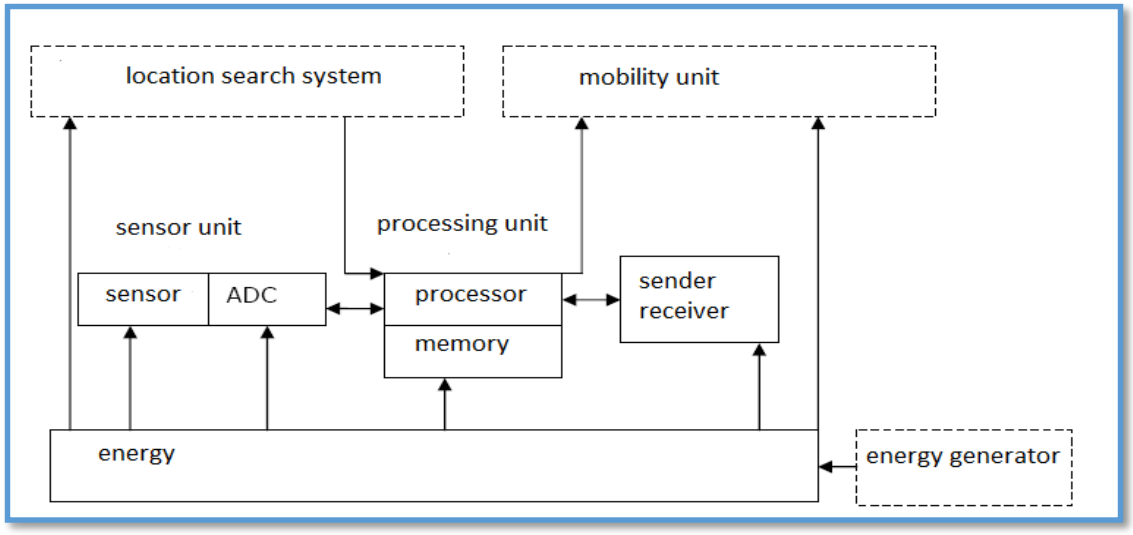

Figure 13 Sensor architecture [18]

❖ **Sensor unit**

It allows the capture of data, i.e. the measurements of physical quantities or analog converters and their conversions into digital data. It is composed of sensor itself and analog-to-digital converters (ADCs). The sensor is responsible for recovering the analog signals that it transmits them to the ADC1 which turns them into digital data and communicate them to The processing unit [38].

❖ **Processing unit**

As the name implies, the processing unit is responsible for all computational operations in a sensor node [10]. It includes a simple processor associated with a limited storage memory, and operates with an operating system dedicated to micro-sensors (TinyOS for example).

❖ **Transmission unit**

This unit is responsible of transmitting or receiving the data packets using a wireless communication device. It is equipped with transmitter / receiver pair; the communication device constitutes a low-range radio antenna in order to save the energy of the sensor nodes [10]. the transmission unit must also contain modulation, demodulation, filtering and multiplexing circuits for radio signal processing.

❖ **Energy unit**

As the main element of the sensor architecture, it supplies all other units with energy. There are essentially two aspects [26]:

- first, to store energy and provide it in the required form
- Second, reconstitute the energy consumed by replenishment by using an external source of the node-sensor such as solar cells.

1.6 WSN features

1.6.1 Deployment Density

The WSN usually consist of a very large number of nodes it is higher than the ad-hoc networks. the number of sensor nodes can reach millions of nodes to ensure a total coverage of the monitored area. In addition, if several sensor nodes are found in one region, one failing node can be replaced by another. However, the deployment density rises the communication challenges for between the nodes. In fact, it causes collisions or damage to the transmitted packets [18].

1.6.2 Limited energy

In a WSN the power supply to each node is provided by a limited energy source, sensor nodes use a tiny size battery as energy resources, which limits their lifespan. the specificity of the WSN applications (military, seismic and other) makes recharging or relocating these batteries a difficult or almost impossible task [27], which leads us to deduce that the lifetime of a node is essentially dependent of the battery life. Thus, the method of managing energy consumption constitutes a major constraint in this type of network.

1.6.3 Scalability

Contrary to traditional wireless networks (personal, local or Extended), an WSN can contain a very large number of sensor nodes (hundreds, Thousands ...). a sensor network is scalable because it has the ability to accept a very large number of nodes that collaborate together to achieve a common goal [22].

1.6.4 Dynamic topology

The instability of the topology of the sensor network is the result of the following three essential factors:

- **Mobility of the nodes**: the sensor nodes can be attached to mobile objects which move freely and arbitrarily, thus introducing an unstable topology of the network.
- **Failure of the nodes**: due to the limited autonomy of the nodes, the topology of the network is not fixed (the nodes that exhaust their energy considered like nonexistent nodes).
- **Addition new nodes:**

New nodes can be easily added, by placing a new sensor in the communication, which contains at least one other already existing sensor node [28].

1.6.5 Self-organization

Self-organization is very necessary for this type of network in order to guarantee its maintenance. Given the different consequences resulting from the instability of the topology

of the sensor network, the latter must be able to self-organize itself to continue its applications [38].

1.6.6 Fault tolerance

The network must be able to maintain its functionality without interruption in case one or more of its sensors failure [28]. This failure can be caused by loss of energy, or by physical damage or environmental interference. Tolerance degree depends on the degree of criticality of the application and the data exchanged.

1.7 WSN Applications domains

WSN can have many applications domains (see the figure 1.4) among them, there are [22,37,10]:

❖ **Military**

Among the WSN characteristics such as fault tolerance, self-organization fast deployment which make this type of network as an appreciable tool in such domain: for example, the WSN can be deployed to controlling the enemies' forces, detection of attacks by weapons: (nuclear, chemical), and control equipment.

❖ **Health**

The WSN can be used in this domain to ensure the monitoring of vital signs and levels of home activity ages or disabilities, remote control of physiological data and it can be also used for monitoring and control of doctors and patients in hospitals.

❖ **Environmental issues**

In order to discovery of natural disasters like: (fire detection, flooding detection…etc.). WSN are deployed in this domain to fulfils them.

❖ **Home**

With the technologies developments, the sensors can be embedded on devices such as: vacuums, microwave ovens, refrigerators control…etc. in order to allow users to control these devices locally or remotely.

❖ **Commercial**

Micro-sensors can be installed in commercial products to control the inventories, vehicles tracking and congestion detection and environmental control in factories and offices.

❖ **Security**

Security represents a very important application domain for WSN. indeed, sensors can be used in buildings in order to detect the alterations in their structure. In addition, a sensor

network can constitute a distributed alarm system, which will be used to detect intrusions across a wide area.

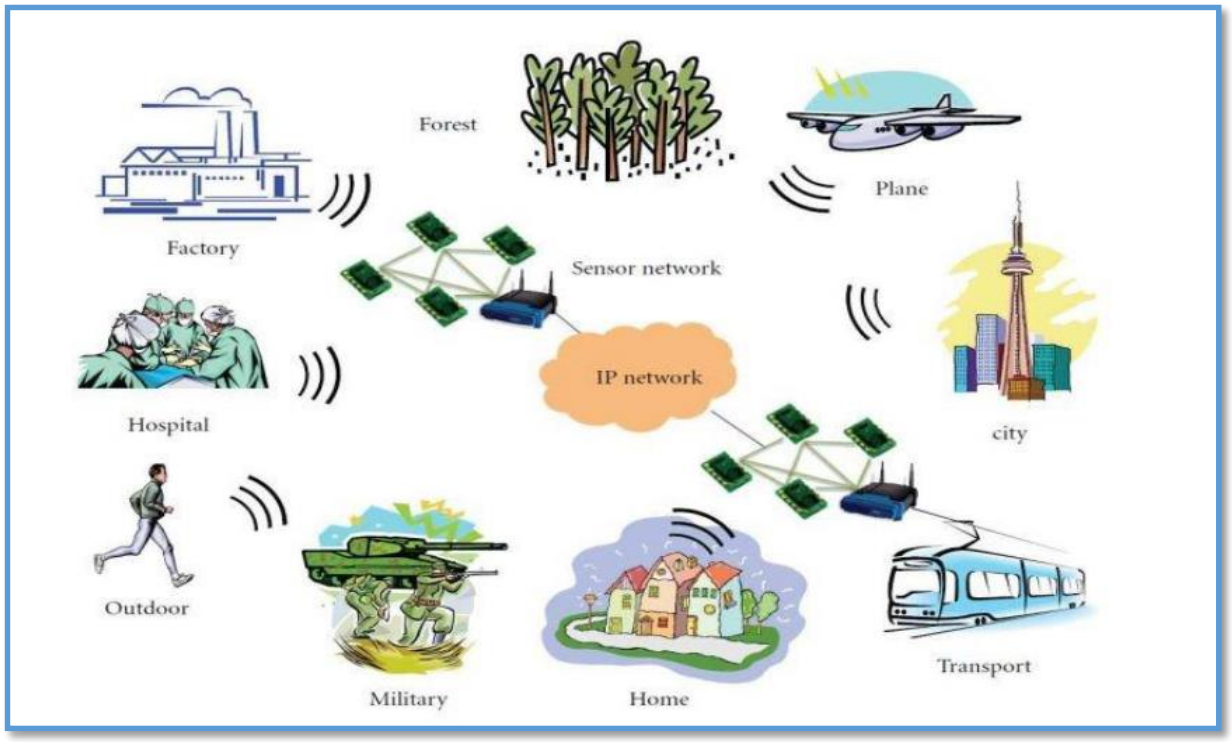

Figure 1.4 WSN application domains [22]

1.8 WSN conception constraints

Research in the field of sensor networks has revealed several problems, in their conception among these issues, we specify [26]:

1.8.1 Fault tolerance

Certain nodes can generate errors due to lack of energy, a physical problem or interference. These problems do not affect the rest of the network; it is the principle of fault tolerance. Fault tolerance is the ability to maintain the functionality of the network without interruptions due to an error on one or more sensors.

1.8.2 Production costs

Often, sensor networks are composed of a very large number of nodes. And these nodes are manufactured products. the price of a node is critical in order to compete with a traditional monitoring network.

1.8.3 Network topology

Due of their high density in the area to be observed, nodes-sensors are able to adapt their operation in order to maintain the desired topology**.**

There are generally three phases in the setting up and evolution of a network [37]:

- Deployment: Either the nodes are distributed in a predetermined or randomly way, they must be organizing themselves autonomously.
- Post-deployment - Operation: during the operation phase, the network topology may be subject to changes due to node position changes or breakdowns.
- Redeployment: The addition of new sensors in an existing network also implies an update of the topology.

1.8.4 Energy consumption

Energy conservation is one of the major issues in the sensors networks. in fact, recharging energy sources is often too costly and sometimes impossible. Therefore, it is necessary that the sensors save as much energy as possible in order to be able to operate [26]. sensor networks operating in a jump routing mode, each node of the network play an important role in the transmission of data. The malfunction of a node involves a change in the topology and requires a reorganization of the network.

1.9 WSN architecture

WSN is composed of a set of sensor nodes [18]. These sensor nodes are generally scattered over a surveillance field in an arbitrary manner, each of these nodes has the ability to collect the data, route them to the sink node (sink / base station), and subsequently to the end user via multi-hop communication. The sink node can communicate with the task coordinator node (administrator) via the Internet or via satellite to analyze these data and take decisions.

❖ **Sink**

Sink or base station it is a particular node in the network and is responsible of collecting data from the different network nodes. Therefore, it must be active and got an unlimited energy.

❖ **Data processing center**

The center which is a collection of all data, sent by the sinks, this place has the role of extracting useful information in order to exploit [23].

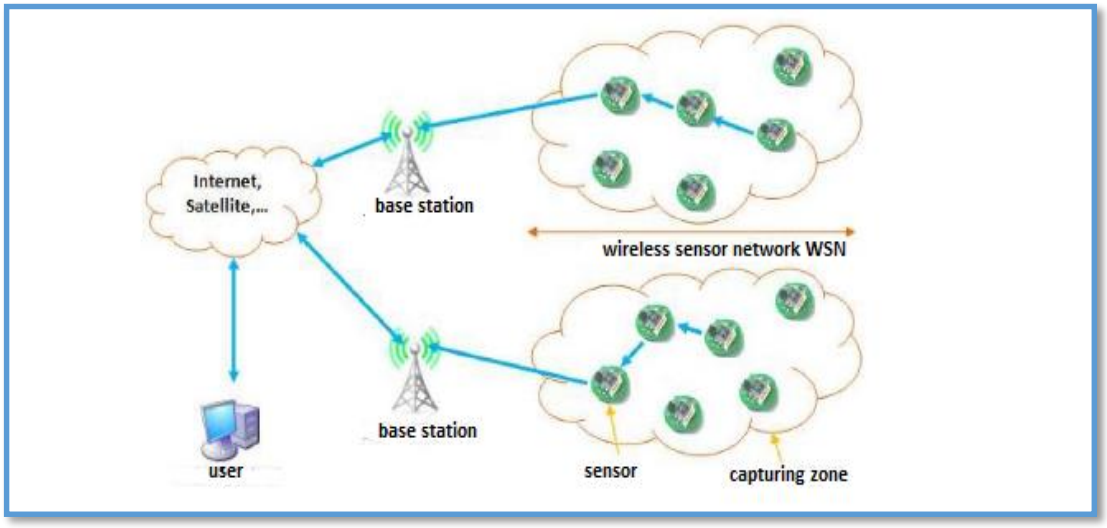

Figure 1.5 WSN architecture [19]

There are two types of architectures for sensor networks: flat sensor networks and hierarchical sensor networks.

1.9.1 Flat sensor networks

Flat wireless sensor network is a homogeneous network, where all nodes are identical in terms of battery and functions, except the Sink, which acts as a gateway and is responsible for the transmission of the collected information to The end user as shown in the next figure

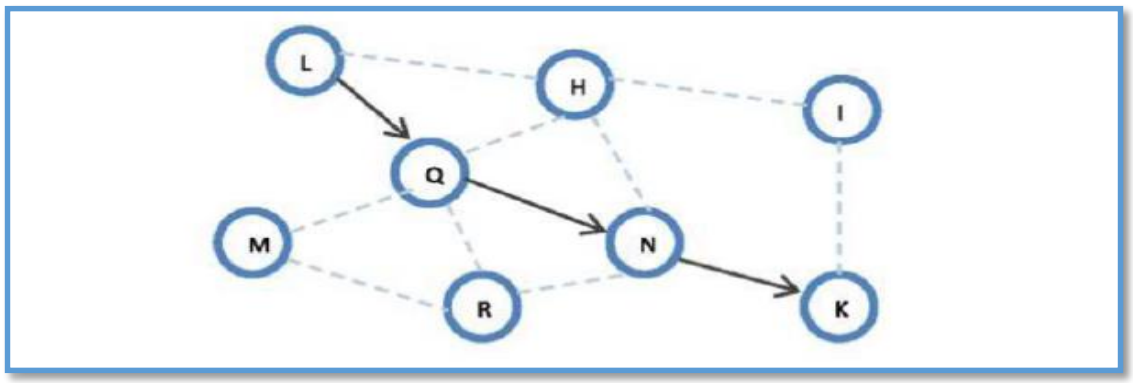

Figure 1.6 Flat routing [28]

1.9.2 Hierarchical sensor network

A hierarchical architecture has been proposed to reduce cost and complexity communications in the WSN [22]. It consists in introducing a set of nodes more costly and more powerful, by creating an infrastructure that discharges the majority of low-cost simple nodes of several network functions.

Hierarchical architecture is composed of several layers: a sensor layer, a transmission layer and an access point layer, the figure 1.7 shows an example of hierarchical architecture

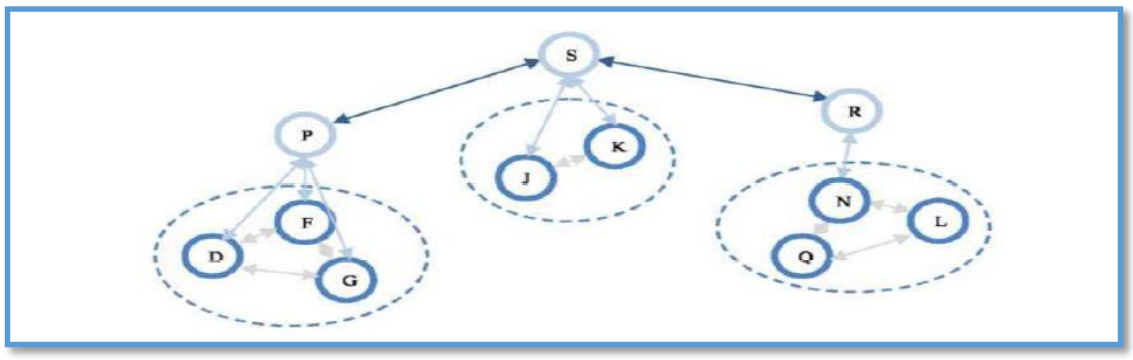

Figure 1.7 Hierarchical routing [28].

1.10 Protocol architecture (protocol stack) [10,27]

Unlike Ad-hoc networks, wireless sensor networks impose additional constraints on communication protocols. Therefore, the traditional layered model (OSI model) does not meet the requirements of this particular type of network. Indeed, the WSN adopt a simplified version of the OSI model, to which are added new layers in order to remedy the constraints and limitations imposed. So, this model comprises 5 layers which have the same functions as the OSI model and three layers for energy management, mobility management and task management.

The goal of a layered system is to separate the problem into different parts (layers) according to their level of abstraction.

Each layer of the model communicates with an adjacent layer (top or bottom). Each layer thus uses the services of the lower layers and supplies them to the higher one.

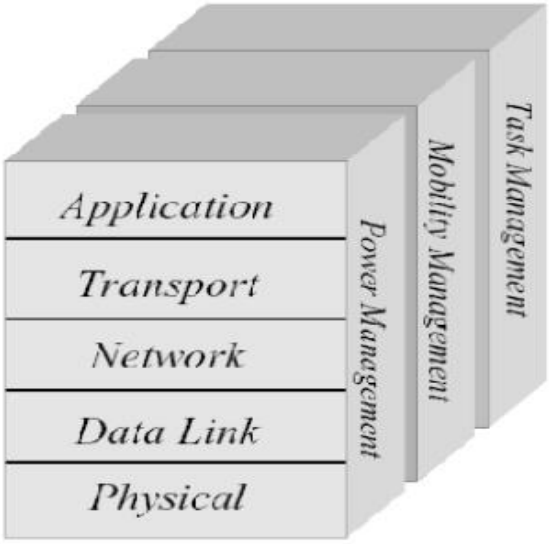

Figure 1.8 Protocol stack architecture [18].

❖ **Physical layer**

This layer is responsible for modulation, signal detection and carrier frequency selection.

❖ **Link layer**

It specifies how the data is sent between two nodes / routers in a distance of a jump. it is responsible for the control of errors, the multiplexing of the data streams, and the control of access to the transmission medium.

❖ **Network layer**

This layer is used to manage the addressing and routing of data, that is to say their routing via the network.

The objective of this layer is to find low-cost routing paths for transmitting the captured data to the base station. Thus, the protocols of this layer must always take into account the resource limitations of the sensor nodes.

❖ **Transport layer**

Its role is the control of the flow, the decoupage the scheduling and the transport of the data packets, and the management of the transmission errors.

❖ **Application layer**

This layer provides an interface for interaction with the human user, whose role is to implement the set of applications and interaction software.

As for the integrated levels in the protocol stack which are: (energy management plane, mobility management plane and tasks management plane) they have the following functions:

- **Energy management plane**

This level consists in managing the energy consumed by the sensors**,** it provides an effective management mechanism to reduce the level of energy consumption and eliminate wasteful sources of energy.

- **Mobility management plane**

This plane is responsible for controlling the movement of the sensor nodes in case where they are mobile. For example, it can record the trajectories of a sensor node in order to help it locate itself.

- **Tasks management plane**

In a sensor network, the nodes can be performing tasks that differ in terms of resource consumption. Thus, a task management plan is often necessary in order to distribute the tasks on the sensor nodes in an equitable way, thus offering an efficient management of available resources.

1.11 Conclusion

Wireless sensor networks are a considerable interest and a new stage in the evolution of information and communication technologies. In this chapter we have presented general concepts related to this type of network, among these concepts we have seen WSN architecture, domains application, conception constraints and more …., also we have identified WSN features that differentiate them from other types of wireless networks. Specific features such as reduced energy consumption, scalability, fault tolerance…etc., which offers infinite possibilities of development in all fields of application.

In the next chapter, we will discuss the concepts of security and their mechanisms in WSN, which is one of the major constraints that confronted the proper functioning of the latter.

CHAPTER 2

SECURITY FOR WSNs

2.1 Introduction

In this chapter, we will give an overview of security in wireless sensor networks. First we present the security objectives and the attacks of different types that can target this particular type of networks. Next, also we will discuss the security mechanism dedicated to the WSN such as: data partitioning, data aggregation, confidence model, and more……finally, we will focus to discuss about cryptography and security protocols which are the main mechanisms in the WSN security.

2.2 Security Objectives

Wireless sensor networks require solutions in many applications that ensure the security of information circling on the network. The security of the information transmitted in the WSN must satisfy a several security conditions [21, 30]:

2.2.1 Data Confidentiality

The network must insure that the transmitted data are confidential and cannot be read by devices or persons other than those entitled to do so. Data must be hidden or encrypted in such a way that no one can access to it. Data confidentiality is paramount in medical applications where patient information must not be disclosed. The same applies to military applications where such information may have some strategic consequences on ongoing actions.

2.2.2 Data Integrity

The integrity of data in any networks signifies that data in that network is valid and undiluted without authorization. This implies that data between the sender and the receiver has not changed in transit by an adversary.

2.2.3 Network Availability

The availability makes possible to specify whether the network is free for the communication of the messages and if the node has the right to use its resources. This requirement ensures the availability of a WSN services, even if one of the internal or external attacks is present as the denial of services attack.

2.2.4 Data Freshness

Data freshness means that the data is recent and it ensures that no enemy replayed old messages. Two types of freshness can be identified. The first is weak freshness.it provides partial message ordering, but carries no delay information. It is required by sensor measurements. The second is strong freshness. It provides an entire order on a request-response pair and allows for delay estimation. It is useful for time synchronization inside the network.

2.2.5 Authentication

The process of authentication of both users and network data is very important in protecting network data integrity and preventing unauthorized entrance to the network. If there are no implemented authenticating mechanisms, attacker can simply access to the network. As result, they can inject dangerous messages without the receivers making sure that the new altered data being used are originates form a malicious source.

2.2.6 Self-Organization

Each node in the WSN should be self-organization and self-healing. after being deployed without further external interventions. This need for self-organization is reflected in the automatic establishment of the encryption keys distribution between the nodes of the network and the management of its keys or else in the development of confidence relations between sensors of the network. To do this, the sensors must have been equipped beforehand with tools that allow them such features.

2.2.7 Secure localization

Often, the utility of a sensor network will be based on its ability to locate each sensor in the network. An array of sensors designed to detect abnormalities will require accurate location information in order to accurately indicate the location of a defect.

2.3 WSN attacks

The WSN are exposed to a variety of attacks because the resource limitations of this type of networks. In this section we describe a list of the most common and known, active or passive attacks that we can find in the WSN [5,17,1].

2.3.1 Eavesdropping attack

It is difficult to detect it because the attacker does not change the data anymore. In this case adversary is only charged by the knowledge of the confidential data or the discovery of the important nodes in the network.

2.3.2 Denial of Service(DOS)

In this class, a wide range of possible attacks is included. These attacks could be carried out by radio interference, but by simply defecting the behavior of a sensor, to prevent the system from functioning properly. In an aggregation scenario, an example of the attacks can be an aggregator that rejects packets received from its member nodes, thus preventing the data from going to the next hop. The selective forwarding attacks considered as an example of these attacks.

2.3.3 Radio jamming

Jamming is a common attack that can be easily done by adversaries by only knowing the wireless transmission frequency used in the WSN. The attacker uses the same frequencies as the WSN which prevents the nodes from communicating because the medium will be saturated by the radio interference.

2.3.4 Black hole attack

A malicious node modifies the routing information in order to force the information to pass itself. Then it creates a black hole in the network and places itself in a strategic routing location then it erases all the messages it should return them, which allows the interrupting of the network routing services in the paths that pass through the node intruder.

2.3.5 Grey hole attack

This attack is a kind of the previous attack, in which only a few type of packets is destroyed by the malicious node. This type of attack is thus more difficult to detect than the attack of black hole because that as long as the malicious sensor behaves in a normal way then it cannot be easily detected.

2.3.6 Wormholes attack

The attack of the wormhole requires the insertion of at least two malicious nodes in the sensors network. These two nodes are connected together by a powerful connection. Which makes it possible to deceive the other nodes over the distances separating them, the figure 2.1 illustrates an example of a wormhole attack:

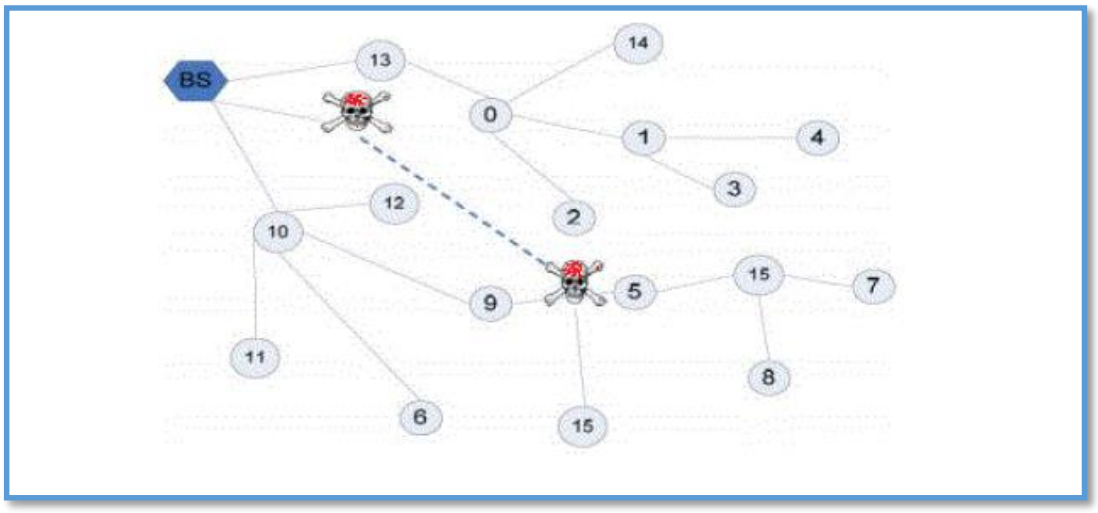

Figure 2.1 Wormhole attack [1]

2.3.7 Sybil attack

In this type, the attacker gets illegally multiple identities on one sensor. It is usually done to fill a neighboring sensor's memory with useless data from non-existing neighbor's sensor.

If a sensor has limit on the number of sensors it keeps data of, it can be used to overwrite useful data.

2.3.8 HELLO flood attack

This attack uses HELLO packets as a weapon to convince the sensors in WSN. In this sort of attack, the attacker with a high radio transmission range and processing power sends HELLO packets to a number of sensor nodes which are dispersed in a large area within neighbor. As consequently, while sending the information to the sink, the victim nodes try to go through the attackers as they know that it is their neighbor and are ultimately spoofed by the attacker (as shown in the figure below).

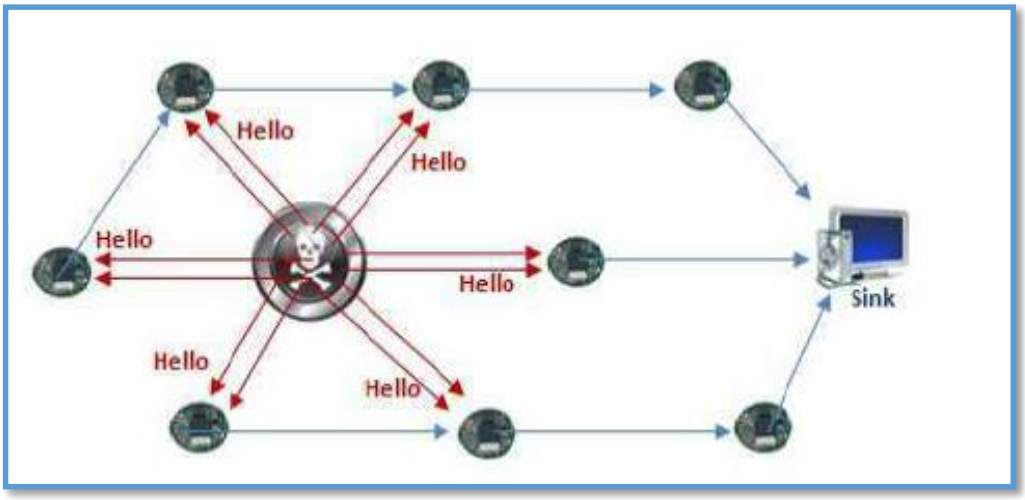

Figure 2.2 HELLO flooding attack [11]

2.4 Security mechanisms for WSN

Several mechanisms are in place to address the issues of security in WSN. We will present some of the most common security solutions in the WSN, there are usual mechanisms which are [27,10]:

- Data partitioning: propose a solution to prevent information retrieval in WSN by partitioning data.
- Data aggregation: reduce the number redundant of messages and consequently reduce energy consumption
- Confidence model: In order to reinforce the level of security, several research proposes security mechanisms, based on the index of trust and reputation. These new mechanisms protect the network from very malicious attacks against which cryptographic solutions cannot do anything.

- Detection Intrusion Systems (SDI): An (SDI) monitors suspicious behaviors and activities of nodes in the network
- Cryptography
- Security protocols.

Cryptography and the protocols of security are the most important mechanisms in the security of the WSN. Which can be summarized as follows:

2.4.1 Cryptography

Cryptography is a security solution that is relatively much safer [36]. The basic idea is to use mathematical methods to transform the original message to a sequence of data that cannot be directly intercepted by a third party (convert information (in clear) into encrypted information, i.e. not comprehensible. It includes a set of techniques that are frequently used in the computer to ensure the confidentiality, integrity and authenticity of data.

There are two types of cryptography encryption, symmetric and symmetric:

2.4.1.1 Symmetric cryptosystems

Symmetric encryption also known as secret key encryption is based on the fact that the same key is shared between two entities to encrypt and decrypt the text.

The sender (ALICE) and the recipient (BOB) must be agree on a key to use before exchanging messages. This key must remain secret [32]. The security of a symmetric algorithm relies on the key: if it is unveiled, then anyone can decrypt Alice's messages.

Figure 2.3 Example of symmetric encryption [32]

Symmetric cryptosystems are divides into two categories: bloc cipher, which encrypts the data bit by bit, and the stream cipher that splits the data into a fixed size blocks. Among the most used cryptosystems we specify the AES cryptosystem as a bloc cipher and the Trivium cryptosystems as a stream cipher:

- **AES (Advanced Encryption Standard)**

Is a block encryption algorithm that requires relatively a little memory, which is resistant to all known attacks particularly differential-linear attacks comparing to DES, which is not resistant to all types of attacks [18], with a simple, design, easily adaptable to 8-bit processors, and low resource requirement.

Recently, AES proposed as a preferable option, as it provides stronger security protection while still being suitable for the low resource requirements on WSN nodes. It can be implemented on WSN nodes. Many AES implementations have been proposed for wireless devices. Also, an example of a performance and energy consumption analysis of different AES implementations on MicaZ sensor node using the TinyOS environment. As mentioned in [12].

- **Trivium**

The Trivium cryptosystem is a hardware oriented synchronous stream cipher which has an internal state of 288 bit registers (s1……., s288) and works with 80-bit secret key using 80-bit initialization vector IV [7]. The process consists of two phases: the first phase is the cipher internal state initialization using both the key and the IV, the second is repeatedly update the state to generate key stream bits.

Depending on the recent works about symmetric cryptosystems of the WSN, we find that the AES and the Trivium cryptosystem are fast and suitable for the sensor node in term of performance: computing time, energy consumption and space memory) [12,20].

2.4.1.2 Asymmetric cryptosystem [12]

Asymmetric encryption also known as public key encryption uses a pair key for encryption: a public key that encrypts data and a corresponding private or secret key for decryption. The private key must remain secret while the public key must be broadcast.

As illustrate the next figure, if the sender (ALICE) want to communicate with the receiver (BOB) in confidential way, he encrypts the message with the public key of BOB, and only BOB able to read the message by decrypt it with his private key.

Figure 2.4 Example of asymmetric encryption [32]

Asymmetric algorithms are based on mathematical problems that are difficult to solve, such as factoring or calculating the discrete logarithm. Considering this difficulty, the security of these algorithms has been proved. Among the best known cryptosystems are: RSA, and several techniques based on cryptography on elliptic curves. We will present them in the next:

❖ **Rivest Shamir Adelman (RSA):** RSA is among the most cryptographic algorithms [29]. The latter is invented by Rivest, Shamir and Adelman in 1978 and it is based on the difficulty of factorizing large numbers. It was so successful that today is the RSA public key algorithm is the most used in the world.

▪ **RSA Encryption and Decryption:**

m is a message in clear and c the cryptogram. The cipher function is a simplified way, $c=m^e \bmod N$ which e is a public key, Because of the relation between e and the private key d, the corresponding decryption function is: $m= c^d \pmod N$.

according to the previous studies related to the WSN, the RSA cryptosystem is requires an important space memory (use keys of long size), it is slow and consumes more energy.so, the RSA is not suitable for WSN.

❖ **Elliptic Curve Cryptography (ECC)**

"Elliptic Curve Cryptosystem" is a public key cryptographic approach based on mathematical aspects of elliptic curves.

An elliptic curve E a prime field F_P can be defined as the set of all tuples $(x, y) € F_P \times F_P$ satisfying an equation of the form [11]:

$$Y^2 = x^3 + ax + b \text{ with } a, b € F_P$$

These tuples are called points with x and y referred to as coordinates. The set of points together with a special point (the so-called point at infinity) forms an Abelian group if the addition is suitably defined.

Two operation are defined over an elliptic curve: addition and doubling of points. given an additional point O= (x, ∞), the so-called “point-at-infinity”, the points on an elliptic curve defined over a finite field form an additive Abelian group. Figure below give a graphical representation of these operations. Note that each group operations requires several operations in the underlying field, so the efficiency and performance of operations over the curve heavily depends on fast implementations of the field operations.

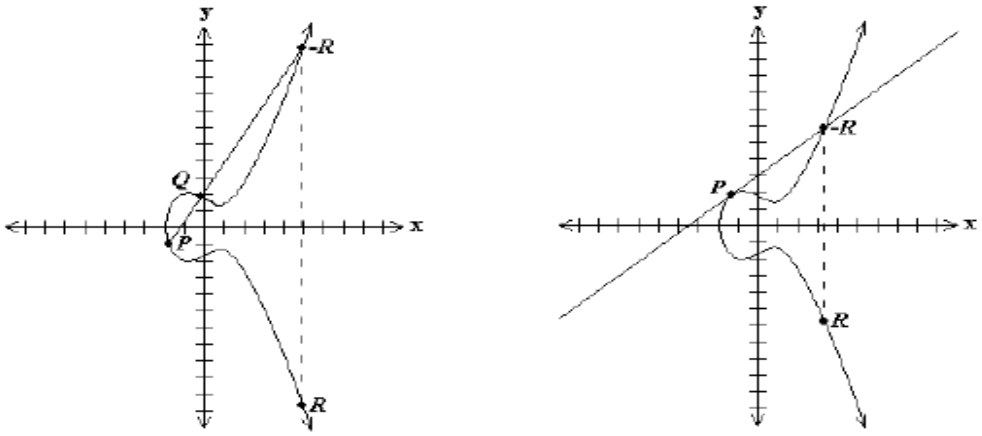

Figure 2.5 Addition and doubling of points operation [11]

R=P+Q R=2P

An elliptic curve domain is completely described by the tuple (q, a, b, P, n, h), where P is a generator of order n for a cyclic subgroup on the curve with shape parameters a and b. q is the order of the underlying field and h the co-factor, that is, the number of points on the curve divided by n. For practical implementations, h should be as close to one as possible.

- **Elliptic Curve Discrete Logarithm Problem (ECDLP)**

The basic building block of public key cryptographic systems is the (assumed) intractability of a hard mathematical problem, like for example the Discrete Logarithm Problem (DLP), that is, given x = ay mod n and knowing x, a and n, it is impossible for an attacker to determine y. The widely used RSA system uses this approach, where a key length of 1,024 bit is assumed to provide a decent level of security. The equivalent to this problem on elliptic curves is the Elliptic Curve Discrete Logarithm Problem (ECDLP), that is, given Q = k.P with Q and P points on the curve and k an arbitrary integer, it is assumed to be impossible to determine k from the knowledge of Q and P. Since until today there is no efficient algorithm available to solve the ECDLP, an ECC cryptographic system with 160 bit is regarded to be at least as safe as an RSA system with 1,024 bit. This makes the use of ECC especially attractive in constrained environments like mobile phones, PDAs, smart cards, embedded devices, and sensor motes.

- **Diffie-Hellman key exchange**

Suppose that two sensors X and Y want to share a common secret but they cannot communicate via a secure channel [8].

The Diffie-Hellman key exchange takes place in the following way:

1. X and Y publicly choose a group G and a point P € G called point generator,
2. X secretly chooses an integer K_x and calculates (k_x*P)
3. Y secretly chooses an integer K_y and calculates (k_y*P)
4. X and Y publicly exchange data (K_x*P) and (K_y* P)
5. X can calculate [K_x*(K_y*p)]
6. Y can calculate [K_y*(K_x*p)]
7. X and Y both have the quantity (K_x*K_y*P) which will be their common secret.

A prospective attacker will have in his possession only the data G, P, K_x*P, K_y*P to find K_X*P. this is called the Diffie-Hellman problem calculator (PCDH).

- **TinyECC**

A configurable library for ECC operations in WSN. the primary objective of TinyECC is to provide a ready-to-use, publicly available software package for ECC based PKC operation that can be flexibly configured and integrated into sensor networks applications. TinyECC implements three standard elliptic curve cryptographic algorithms such as: Elliptic Curve Diffie Hellman (ECDH) key agreement, Elliptic Curve Digital Signature Algorithm (ECDSA), and Elliptic Curve Integrated Encryption Scheme (ECIES).

From the proposed works which have been designed for development of TinyECC the experimental evaluation of TinyECC on several common sensor platforms, including MICAz, Tmote Sky, and Imote2.as mentioned in [23].

❖ **RSA and ECC Comparison**

This table shows a comparison of key size between RSA and ECC on asymmetric cryptography:

RSA key size	ECC key size
1024	160
2048	224
3072	256
8192	384
15360	512

Table 2.1 Comparison of key size in bits between RSA and ECC [17]

Depending on what is stated in the table, we note that cryptography on elliptic curves allow the use of medium size keys compared to RSA while providing the same performance.as a result, the advantage of ECC is the use of shorter key than other cryptographic methods that allow faster execution the operation of encryption (decryption) and requires a less need in memory. This observation makes it possible to favor the use of elliptic curves for sensor node systems with limited resources in terms of memory and CPU.

2.4.2 Digital Signature

The digital signature is a cryptographic system ensuring the non-repudiation of the source (non-repudiation is the guarantee that someone cannot deny having sent a message once it has been signed with their digital signature). It is based on asymmetric keys [16]. The transmitter (A) signs the data to be transmitted with its private key (A) by producing a digital signature (1). The latter is then sent with the data (2). If it can be decrypted with the public key (A) by the receiver (B) and if its result is identical to the data received, then the signature is valid (4), that means, the data come from their legitimate sender, who will not be able to deny the emission of these data in the future.as illustrated in the figure 2.6.

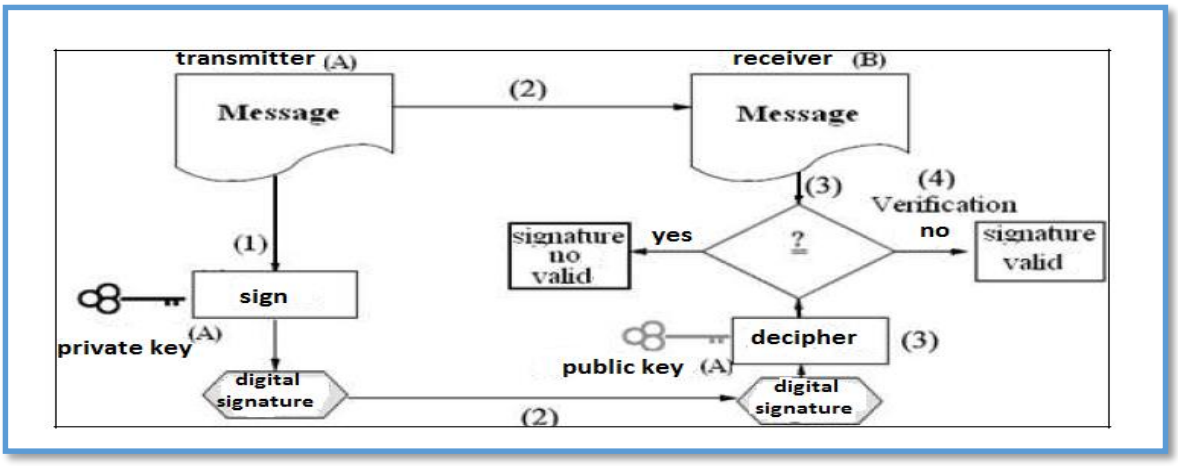

Figure 2.6 Digital signature [16]

2.4.3 Hash Function

The hash function serves to calculate a short fingerprint of fixed size from an input a data of arbitrary size. This size generally varies between 128 and 515 bits. The fingerprint, also known as condensed or simply chopped, must depend on all bits of the message and are used to present the message in a compact way [35]. A hush function is an entirely public algorithm and no secret value occurs at any point in the calculation. Hush function is mainly used to: insure integrity, build cryptographically safe random generators, for the theoretical modeling of one-way function such as the random oracle model in proofs of safety.

A hush function H is a function which takes as input a random size data m, and outputs a fixed size condenser n.

$$H: \{0,1\}^{*} \rightarrow \{0, 1\}^{n}$$

$$m \rightarrow H(m)$$

The imprint of a data, serves to represent it and makes it easy to identify it. It is difficult to find two random messages that give the same fingerprint and this leads to collisions resistance. This fingerprint is recalculated by the receiver to compare it with that calculated by the transmitter if they are different then the data has been altered during their transmission.

Many of research works on security related implementations of hashes for WSN. It shows that, due to their reduced complexity, they constitute the prudent energy efficient option compared to encryption techniques. Among them we recall a one-way hash-function for public key authentication. proposed by Du et al, and an authentication and key establishment scheme that is an energy efficient and especially suited to sensor networks. Proposed by Delgado-Mohatar et al.as mentioned in [14].

❖ **Message Authentication Codes (MAC)**

These cryptographic mechanisms allow both to ensure the integrity of the message sent and to authenticate its sender. To achieve this; when sending the message, the authentication code is added to the message [11]. A MAC can be constructed from a hush function. If X is a message then to guarantee the integrity of X it sends or stores the pair (x, h (x, k)) where h (x, k) is the fingerprint of x via a hush function h using a key k. The message is considered integral if its imprint that cannot be falsified without knowing k (knowing x and h (x, k) accompanies it.

2.4.4 Security protocols for WSN

Security is broadly used term encompassing the characteristics of confidentiality, authentication, integrity, freshness, and …. etc., for this reason, variety protocols of security have been developed for wireless sensor network. We classify the main aspect of these protocols into two major categories: authentication protocols and authenticated key exchange protocols.

❖ **Authentication protocols**

Several protocols have been proposed to ensure authentication in WSN. Most of these protocols adopt authentication mechanisms based on the generation and exchange of secrets codes (MAC). Among the authentication protocols there are: SPINS, TinySec and MiniSec [31,2]:

SPINS (Security Protocols for Sensor Networks) developed by Perrig, consists of a set of security protocols that acts through encryption and message authentication codes it offers two kind of protocol SNEP and μTESLA for confidentiality, data integrity, data authentication, freshness of weak message, and protection of replay message. The TinySec (Tiny Security) was designed and implemented in the TinyOS operating system to be a mechanism for providing confidentiality, integrity and authenticity of the data link layer. It uses the CBC mode of operation that may be combined with various block ciphers as RC5 and skipjack. The MiniSec (Mini Security) is a protocol layer of WSN security that uses OCB (Offset Codebook) mode to operate the block cipher, which eliminates the need of adding filler to the clear text blocks.

❖ **Authenticated Key exchange protocols**

The key exchange protocols allow two sensor node to jointly establish a secret key over an insecure channel without any prior knowledge of each other [24]. The exchange key can be used for achieving various cryptographic primitives like authentication, MAC, integrity using symmetric cryptography. Among the key exchange protocol, we specify the Elliptic Curve Diffie Hellman(ECDH). we will discuss about this protocol in the next chapter.

2.5 Conclusion

Wireless sensor networks(WSNs) security is a very important issue especially with the strict constraints of computational power, energy and communication medium of sensor nodes. In this chapter, we discussed the security aspects of WSN and the different attacks of this particular type of networks, which imposes additional constraints that makes the network security a challenge. We have also introduced the different mechanism of security focusing on the cryptography and security protocols (authentication protocols, and key exchange protocols).

The next chapter will be devoted to the presentation of different security protocols.

CHAPTER 3

SECURITY PROTOCOLS FOR WSNS

3.1 Introduction

WSNs are special type of network which has many constraints such as low power supplies, small size memory and short time execution, due to these constraints it is difficult to employ the existing security approaches in the WSN. Therefore, security protocols for WSNs are focused on the conquest of these constraints.

In this chapter, we describe the security protocols for WSN of different categories such as key exchange protocol ECDH, encrypted key exchange EKE, ECDSA which based on the digital signature and the new proposed protocol that we have focused to implement in our work.

3.2 Notations

First, we start to define the used notations as follows:

Symbol	Notation
E	Elliptic curve
F_p	A finite field
P	A prime number
$H()$	Hush function
K	An integer number
D	A private key
Q	A public key
G	Generator point of order n
X	Random number chosen by A
Y	Random number chosen by B
ID_A	Node identity of A^{th} node
ID_B	Node identity of B^{th} node
BS	Base station
S_b	Random number chosen by BS, to act as private key
P_b	Public key of BS , $P_k = S_{b.G}$
R_A	Private key chosen by A^{th} node
P_A	Public key of A^{th} node
R_B	Private key chosen by B^{th} node
P_B	Public key of B^{th} node
S_A	Signature of A^{th} node
S_B	Signature of B^{th} node

R_1	Random number chosen by Ath node
R_2	Random number chosen by Bth node
MK	Master key

3.3 Encrypted Key Exchange (EKE)

Encrypted Key Exchange(EKE) protocol [39]. This method provides mutual authentication through the use of a short password. And no need for public key certificate.

EKE can be used with a variety of asymmetric cryptosystem and public key distribution system. It works especially well, when it comes to expential key exchange in the WSNs.

Several EKE protocols has proposed for WSNs such as Diffie Hellman base EKE protocol (DH_EKE) [33].

In this section we define the use of this protocol in WSN. Such as the messages sent between two sensor node for the purpose of authentication and key derivation. Let these nodes be A and B, they must perform the following steps:

- From A to B: A. {exp (g, x)} _k (A, B), which k (A, B) is a password (the sherd key between A and B). then, B computes master key MK such that MK= H (A, B, exp (g, x), exp (g, y), exp (g, xy))
- From B to A: B. {exp (g, y)} _k (A, B), H(MK,1), A computes master key KM
- From A to B H(MK,2)
- Session key K= H(MK,0),

The figure bellow shows the authentication and key exchange process between A and B

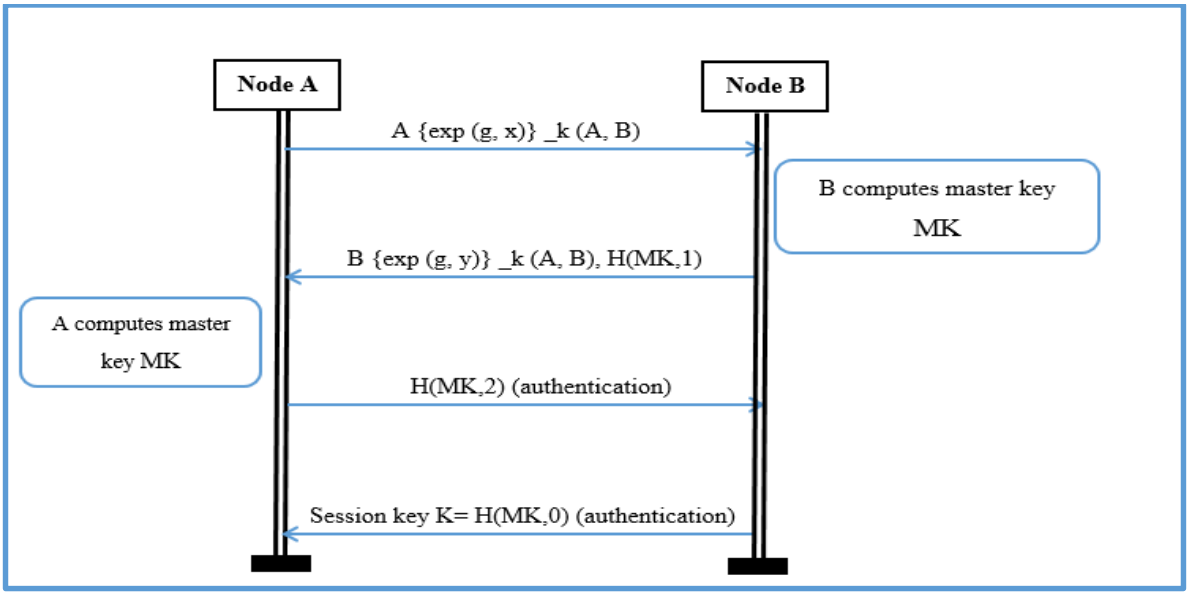

Figure 3.1 Mutual authentication and key exchange

3.4 Elliptic curve Diffie Hellman (ECDH) Protocol

The Elliptic Curve Diffie Hellman (ECDH) [3], distinct from the general Diffie Hellman (DH) in the way that it is based on the Elliptic Curve Discrete Logarithm Problem (ECDLP) instead of the Discrete Logarithm Problem(DLP) as described in the chapter 2, ECDH is an anonymous key agreement protocol which allow two parties to establish a shared secret key over an insecure channel where each of the parties have an elliptic curve public-private key pair. Suppose Alice and Bob agree on the elliptic curve group E of order n and a primitive element P in E, which then also has the order n. E, n and P are assumed to be known to the adversary. The ECDLP, which the ECDH is based on, is defined as the computation of the integer k given P and Q such that Q = [k]P. The ECDH allows to node A and node B compute a shared secret key, using the property of the ECDLP.

Suppose A need to set up a mutual key with B, however the channel accessible to them might be eavesdropped by a third party. G is the generator point of n order on the chosen elliptic curve is the chosen secret of A and y is the chosen secret of B.

In order to establish a shared secret key, A and B must follow the following operations:

1. A chooses a random number x € [2, n-1]
2. Computes the scalar multiplication Q= x.G (as this is an ECDLP the secret x cannot be extracted from x.G)
3. Send Q to B
4. Similarly, B choose a random number y and computes R=y.G and send R to A
5. A and B receives Q and R respectively, and compute the shared secret key

S= x.G = y.G = x.y.G

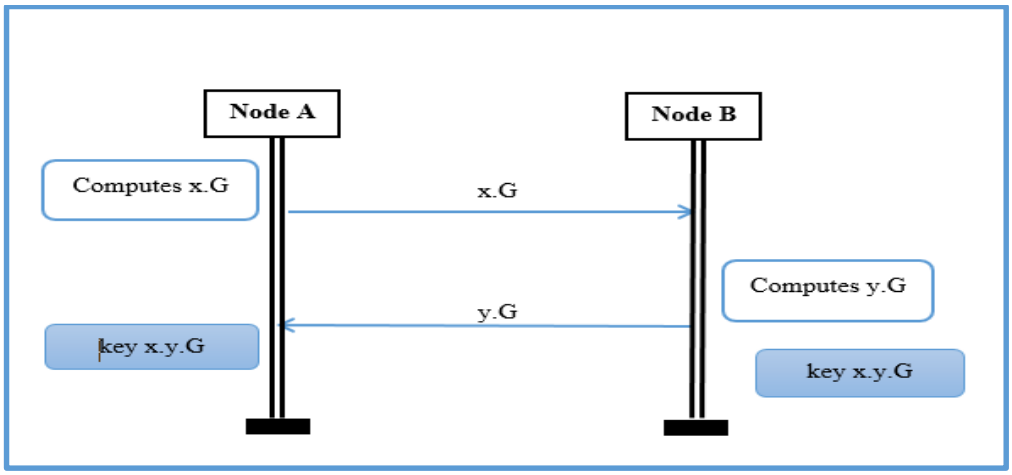

Figure 3.2 Elliptic Curve Diffie Hellman key exchange method [3]

3.4.1 Man in the Middle Attack (MIMA)in ECDH

The ECDH is also concerned with other types of attacks than finding the shared secret key S. One of these is the man-in-the-middle attack(MIMA), a man-in-the-middle attack is an attack where the attacker secretly relays and possibly alters the communication between two sensor node [25], while they believe they are directly communicating with each other. A third party E, who is attacking, E chooses two random numbers p and q and manipulates the sequences in such a way that a separate pair of key is established between A and E where in A believes it has established a key with B and Similarly between B and E where in B thinks that it has established a key with A, as illustrated in the next figure.

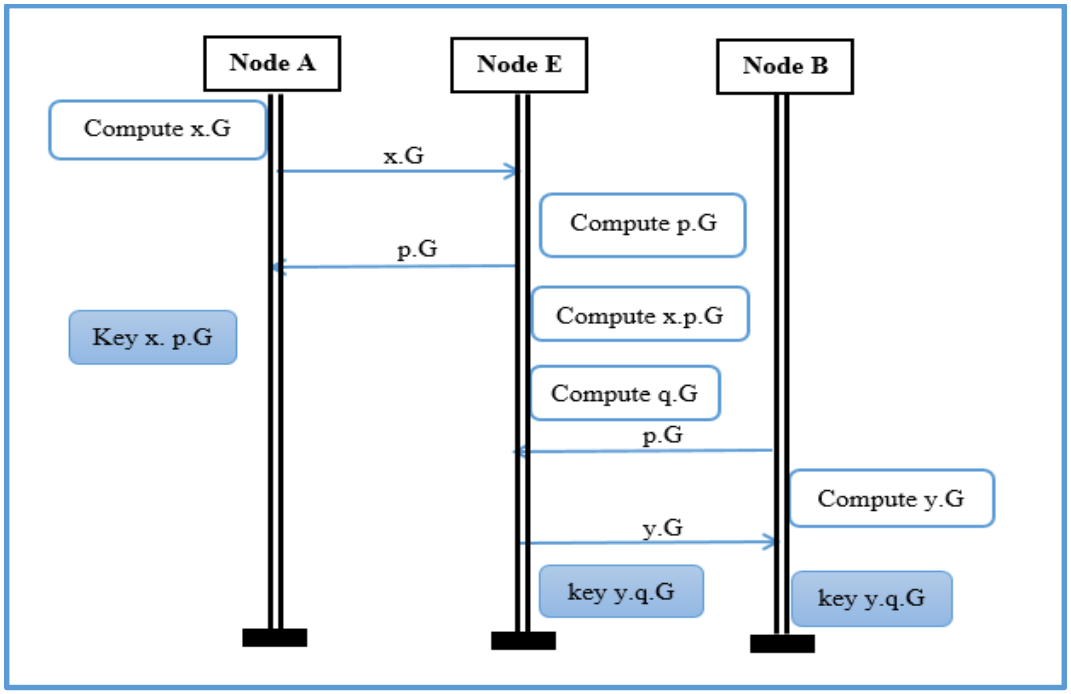

Figure 3.3 Man-in-The-Middle Attack in ECDH [3]

3.5 Elliptic Curve Digital Signature Algorithm (ECDSA)

ECDSA Protocol is proposed by Johnson et al [15]. It offers a variant of the digital signature algorithm (DSA) which use elliptic curve cryptography and serves the same purpose as key generation, signature generation and signature verification, ECDSA comprises of the following phases:

1. Signature generation:

A negotiates with B and select a point G, Alice chooses parameters a, b, p, n to construct the elliptic curve E_p (a, b) and then A obtains a private key d and a public key Q=d.G. Suppose a message m, in order to A can generates the signature for the message m he should do the following steps:

1. Selects a random key k in [1, n-1]
2. Calculates a curve point k.G=(x_1,y_1)
3. calculates r = $x_1$mod n. if r=0, go back to step 1
4. Calculates e= H(m), where H is a hush function
5. Calculates s =k^{-1}(e + dr) mod n. If s = 0, then go back to step 1).
6. Send the digital signature (r, s) and m

2. Signature verification:

To verify A's message and signature (r, s), B should do the following;

1. Verify that r and s are integers in [1, n-1]. if anyone is not, the signature is invalid
2. Calculates e= H(m),
3. Calculates w= s^{-1} mod n
4. Calculates u_1 = (e×w) mod n, u_2 = (r.x) mod n
5. Calculates the curve point X= u_1G + u_2Q = (x_1, y_1)
6. If X= 0, the signature is invalid, and refuse it, else calculate v= x_1 mod n
7. B will accept the signature if and only if v=r

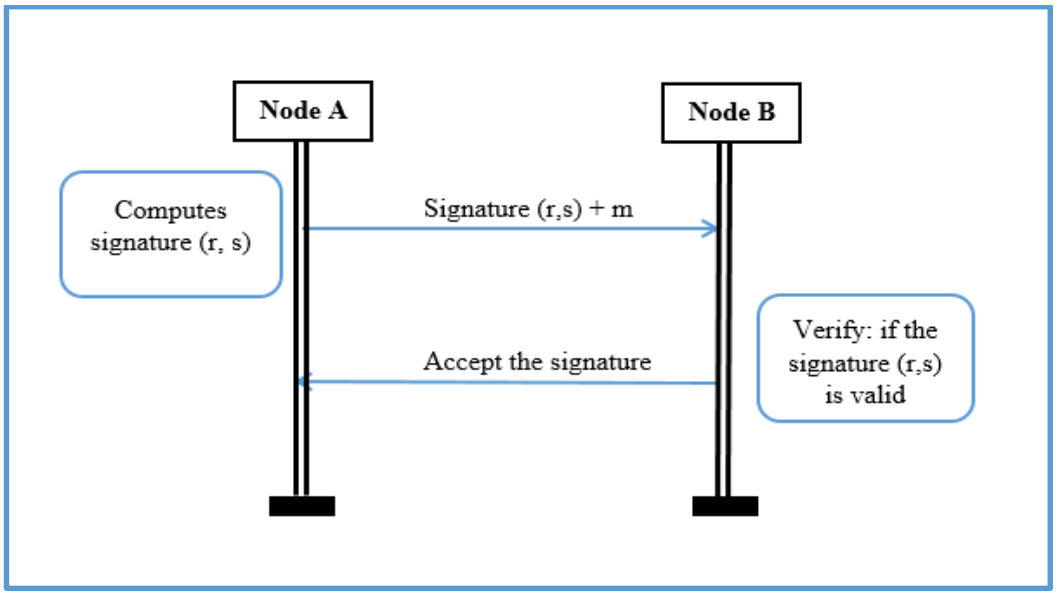

Figure 3.4 Signature verification in ECDSA

3.6 Protocol of Moon et al.

The use of ECDSA is not suitable for achieving mutual authentication between the entities like the base station, cluster heads and nodes. Based on the study of ECDSA, Recently, Moon et al. [4] proposed a mutual entity authentication protocol with the help of a computationally low signature scheme comprises of four phases.

❖ **Cryptographic primitives**

The proposed protocol based on two asymmetric cryptographic primitives:

- hush function H ()
- digital signature algorithm (DSA)

❖ **proposed node authentication protocol**

The protocol designed comprises of the following phases:

1. Initialization phase

Base station functions like a key generation and distribution center and set up the system parameters as the following:

The parameters group of the system is D = (Fp, E, G, n, h, H, Sb, Ps)

2. Signature generation phase

In this phase the base station performs the following steps:

- Selects a random number S_b in [1, n-1] as its private key
- Computes its public key $P_p = G. Sb$
- Generate a random number r such that $1 < r < n-1$ for each node A
- Computes public key for each node $P_A = G. r_A$
- Computes a signature for each node using the signature function s_A (r_A, Id, s_b). such that: $s_A = r_A^{-1} [r_A .h (id)+s_b]$

Each node is preloaded its specific signature s_i, public certificated of base station s_b and public certificate of each node P_A is made public to the network

3. Signature verification phase

It's a check-self by the node to ensure that the signature s_i pertains to it and has been correctly generated and thereafter loaded into it. The node uses its identification, public certificate, and the public certificate of the base station to verify the authencity of its signature as the following:

- Node A verifies its signature value by computing the RHS value of this equation $S_A = r_A^{-1} [r_A .h(Id) + s_b]$. and comparing it with the LHS, if this equation is satisfied then the signature s_A of node A is verified. (Note that the steps to be initiated by node A are indicated at LHS and the steps to be initiated by node B are indicated at RHS)
- $R_A = s_A^{-1}[r_A .h(Id) + s_b]$: multiplying each side by G
- $G. r_A = G.^{s}{}_A^{-1}[r_A .h(Id) + s_b] = s_A^{-1}. Gr_A .h(Id) + s_A^{-1} .G.sb$
- $G.r_A = s_A^{-1}.h(Id)$. public certificate of node A + s_A^{-1} public certificate of base station

4. Mutual authentication phase

In this phase the nodes A and B have to authenticate. This is called the mutual authentication between them which can be achieved by using their signatures, there public certificate, and the public certificate of the base station. As shown in the figure bellow

The process of the mutual authentication between A and B is as the following steps:

❖ Node A:

- Step one: selects a random number R_1 and compute G. R_1
- Step two: computes s_A (G.R_1 + r_A.G), s_A.G,G.R2 Send it to node B
- Step five: verify s_B (G. R_2 + r_B. G) = (s_B. G). R_2 + s_B (r_B. G), if LHS =RHS then node A authenticate node B

❖ Node B:

- Step one: Selects a random number R_2 and compute and send G. R_2 to node A
- Step three: verify s_A(G. R_1 + r_A. G) = (s_A. G). R_1 + s_A (r_A. G), if LHS =RHS then node A authenticate node B
- Step fourth: compute s_B (G. R_2 + r_B G), s_B G then send it to node A

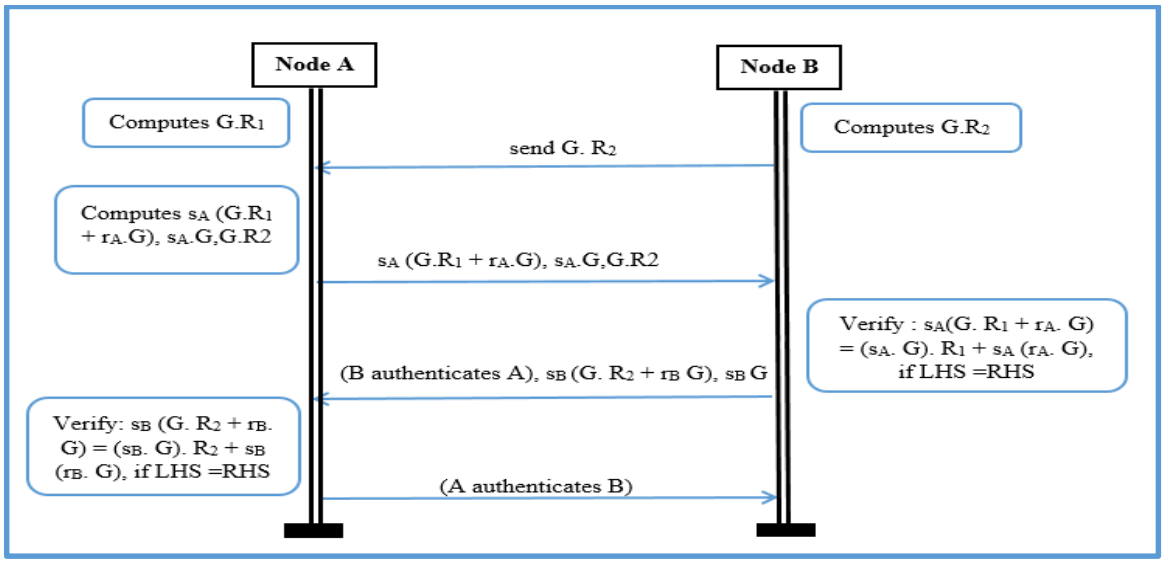

Figure 3.5 Node mutual authentication protocol

3.7 Conclusion

Chapter 3 dealt mainly with security protocols for WSNs description with different categories. The next chapter will be devoted to the performance evaluation of these protocols using a simulation realized with two type of simulators which are TOSSIM and Avrora.

CHAPTER 04

IMPLEMENTATION AND SIMULATION

4.1 Introduction

For a better development results of our work we choose to use the Micaz sensor and the TinyOS as simulation environment with the NesC language. In this chapter, we display a simulation of the security protocols described in the previous chapter and experimental results obtained of their evaluation of performance.

4.2 Experimental environnement

4.2.1 TinyOS operating system

TinyOS is an event-driven operating system designed for sensor networks it is implemented entirely in NesC and respects an architecture based on the combination of components, allowing them to reduce the size of the code necessary to its implementation [46]. It occupies a very small memory space in its minimum distribution (512 bytes). Therefore, it adapts to sensors with very limited memory resource.

However, the TinyOS components library is particularly complete since it includes network protocols, sensor drivers and data acquisition tools

4.2.1.1 TinyOS caractiristics [9]

- Competition: uses event driven architecture
- Modularity: application composed of components and application compiled into a single executable
- Communication: use model event/command, non-Preemptive FIFO Scheduling
- No kernel / user separation

Figure 4.1 TinyOS logo [9]

4.2.2 NesC Language

Is a component-based, event-driven programming language used to build application for the TinyOS platform [41]. NesC is built as an extension to the C programming language with components "wired" together to run application on TinyOS. The name NesC is an abbreviation of "network embedded systems C".

4.2.2.1 Components and interfaces

❖ **Component**

NesC programs are built out of components, which are assembled (wired) to form whole programs. There are two types of components [41,42]:

1. Configuration: used to assemble other components, Linking the interfaces used by components to the interfaces offered by Other components. This is called "wiring".

```
configuration name { /**  provided interfaces*/  }

implementation {/** list of modules and configurations used */
                 main components , module....,configuration ....,
                 /** description of links */
                provided interface -> requiered interface

}
```

Figure 4.2 Configuration syntax

2. Module: The application code works by implementing one or more interfaces.

```
module Name {
  provides { /** list of provided interfaces
              Interface nameInterface */            }

  uses { /** list of required interfaces
              Interface nameInterface */            }

  }

implementation { /** variable declaration
                  implementation of the functions described by the provided interfaces */
  } }
```

Figure 4.3 Module syntax

❖ **Interface**

Are bidirectional, they specify a set of function to be implemented by the interface's provider (commands) and a set to be implemented by the interface's user (events).

```
interface name {
        //list of command
        command datatype name(datatype arg1);
        ....
        //list of events
        event datatype name(datatype arg1);
        ....
}
```

Figure 4.4 Interface syntax

4.2.2.2 Naming convention in NesC

- file extension is: .nc
- Clock.nc: an interface (or configuration)
- ClockC.nc: a configuration
- ClockM.nc: a module

4.2.3 Cryptographic library

In order to evaluate the cryptographic primitives with the TOSSIM and avrora simulator we have used the TinyECC library, which is based on NesC language, this library provides digital signature schema (ECDSA) and the key exchange protocol (ECDH) [43].

4.3 Simulation Tools

WSNs Simulation is very important for WSN development protocols, schemes. in this section we illustrate our simulation tools used which are: TOSSIM and Avrora.

4.3.1 TOSSIM

TOSSIM is an open source operating system targeting embedded operating systems [44], specifically designed for WSN running on TinyOS. TOSSIM is a bit-level discrete event network built in Python, a high-level programming language emphasizing code readability, and C++, It's used with a graphical interface JTOSSIM for a better understanding and visualization of the network state. It's can be run on Linux Operating Systems or on Cygwin on Windows.

❖ **TOSSIM limitations**

- Designed [45] to simulate behaviors and applications of TinyOS, and it is not designed to simulate the performance metrics.
- Cannot correctly simulate issues of the energy consumption and time execution

- Only compatible with TinyOS

4.3.2 PowerTOSSIMz

TOSSIM does not have the ability to check the rate of energy dissipated during application execution. However, the need to verify energy consumption in a WSN is of primary interest. PowerTOSSIM is a new simulator integrated in TOSSIM, which allows generating a file of the extension [46]. Trace which records the details of the simulation like the energy consumed in the network.

4.3.3 AvroraZ

Avrora is a cycle-accurate instruction-level simulator specifically designed for WSN [47], implemented in Java. was developed by University of California Los Angeles Compilers Group. It simule all the instructions executed in the sensor node and support energy consumption simulation.

❖ **Avrora merits [45]**

- The codes in avrora run instruction by instruction which provide faster speed and better scalability
- Can support thousands of nodes simulation
- Can simulate different programming code projects
- Avrora provides more accuracy than TOSSIM

4.4 Evaluation methods

We have evaluated the performance with the TOSSIM and AvroraZ simulators by using three performance metrics which are energy consumption, execution time and required memory, through the following steps:

1. execution time: calculate the execution time by using the commands:
   - opt/tinyos-2.x/apps cd testECDSA
   - opt/tinyos-2.x/apps/testECDSA cd build
   - cd micaz
   - cp main.exe ecdsa.elf
   - alias avrora = java –jar /opt/avroraZ/avroraZ.jar
   - avrora -platform=micaz –monitors=leds –second=5.0 –report-second ecdsa.elf
2. energy consumption: calculate the energy consumption by using the commands:
   - opt/tinyos-2.x/apps cd testECDSA
   - opt/tinyos-2.x/apps/testECDSA cd build
   - cd micaz

- cp main.exe ecdsa.elf
- alias avrora = java –jar /opt/avroraZ/avroraZ.jar
- avrora -platform=micaz –monitors=energy –second=5.0 ecdsa.elf

3. memory size: calculate the memory size by using the commands:
   - opt/tinyos-2.x/apps cd testECDSA
   - make micaz

4.5 Experimental results

To make a comparative study between the ECDSA, ECDH, and Moon protocol, we relied on the performance evaluation of our selected sensor type, which is the Micaz in terms of RAM and ROM occupation, energy consumption and time execution by using TOSSIM and avrora simulator

4.5.1 ECDH

❖ **Required memory**

The figure 4.5 shows the results obtained after the compilation of the ECDH protocol in terms of required memory

```
    compiling testECDH to a micaz binary
ncc -o build/micaz/main.exe  -Os -fnesc-separator=__ -Wall -Wshadow -Wnesc-all
NED_TOS_AM_GROUP=0x22 --param max-inline-insns-single=100000 -DSECP160R1 -DBARR
N   -DIDENT_APPNAME=\"testECDH\" -DIDENT_USERNAME=\"wcu\" -DIDENT_HOSTNAME=\"wc
x09948d3bL -fnesc-dump=wiring -fnesc-dump='interfaces(!abstract())' -fnesc-dump
stECDH.nc -lm
/opt/tinyos-2.x/tos/chips/cc2420/lpl/DummyLplC.nc:39:2: warning: #warning "***
    compiled testECDH to build/micaz/main.exe
          26352 bytes in ROM
           2394 bytes in RAM
avr-objcopy --output-target=srec build/micaz/main.exe build/micaz/main.srec
avr-objcopy --output-target=ihex build/micaz/main.exe build/micaz/main.ihex
    writing TOS image
```

Figure 4.5 Evaluation of required memory in ECDH

❖ **Energy consumption**

The figure below presents the results obtained by the compilation of ECDH in term of energy consumption

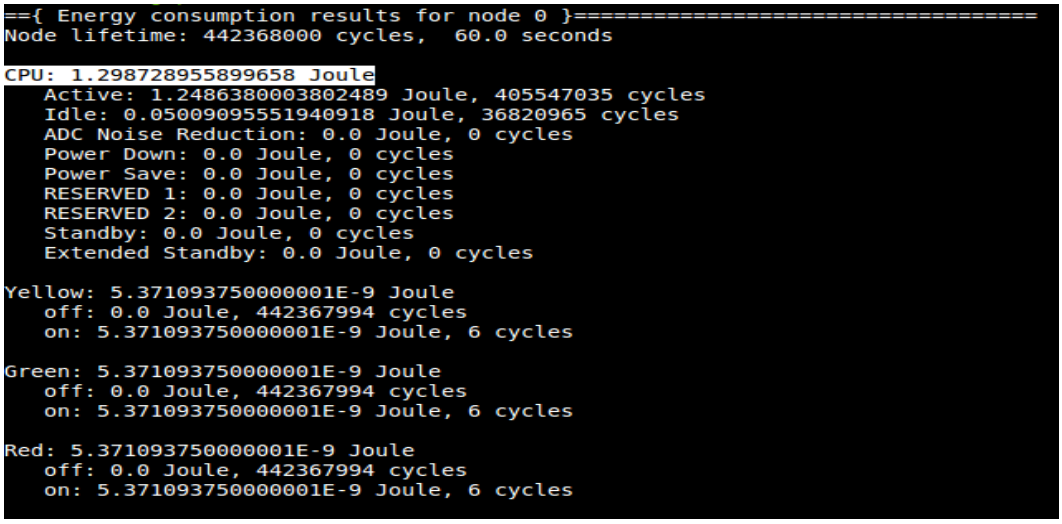

Figure 4.6 Evaluation of energy consumption in ECDH

❖ **Execution time**

The next figure shows the execution time evaluation in the ECDH after the compilation

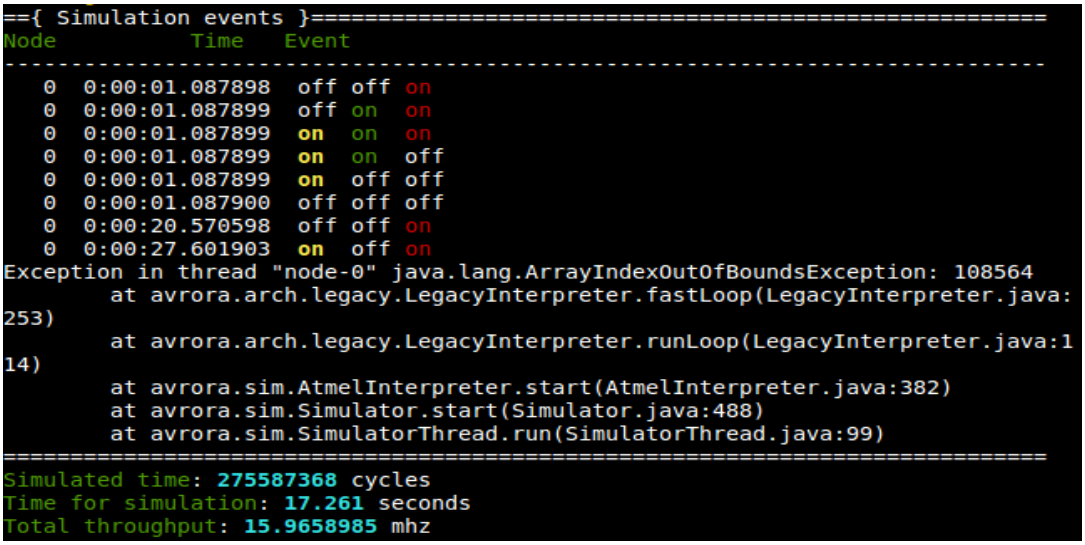

Figure 4.7 Evaluation of execution time in ECDH

4.5.2 ECDSA

❖ **Required memory**

The next figures show the results obtained after the compilation of the ECDSA protocol for two phases:

- Signature phase

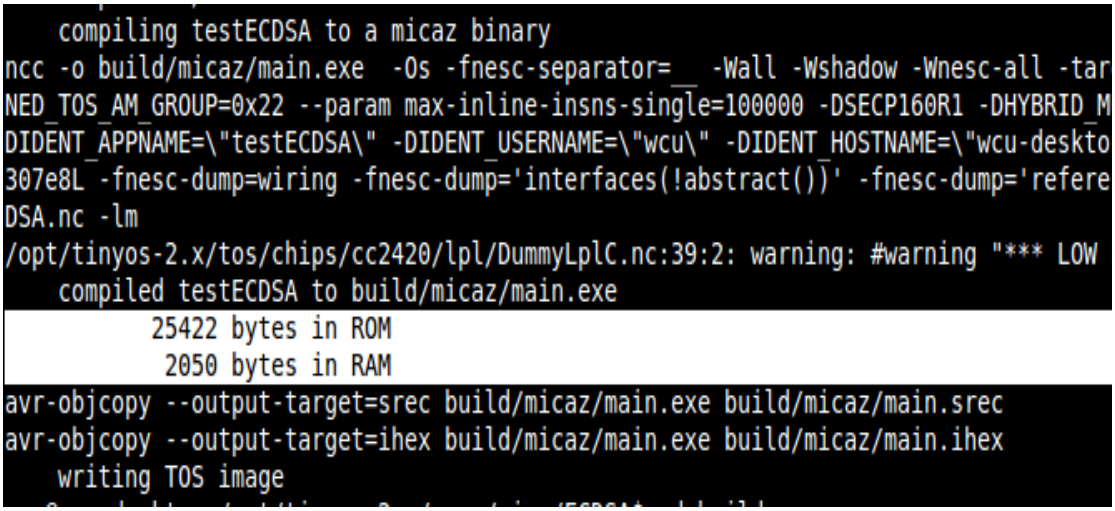

Figure 4.8 Evaluation of required memory in ECDSA

- Verification phase

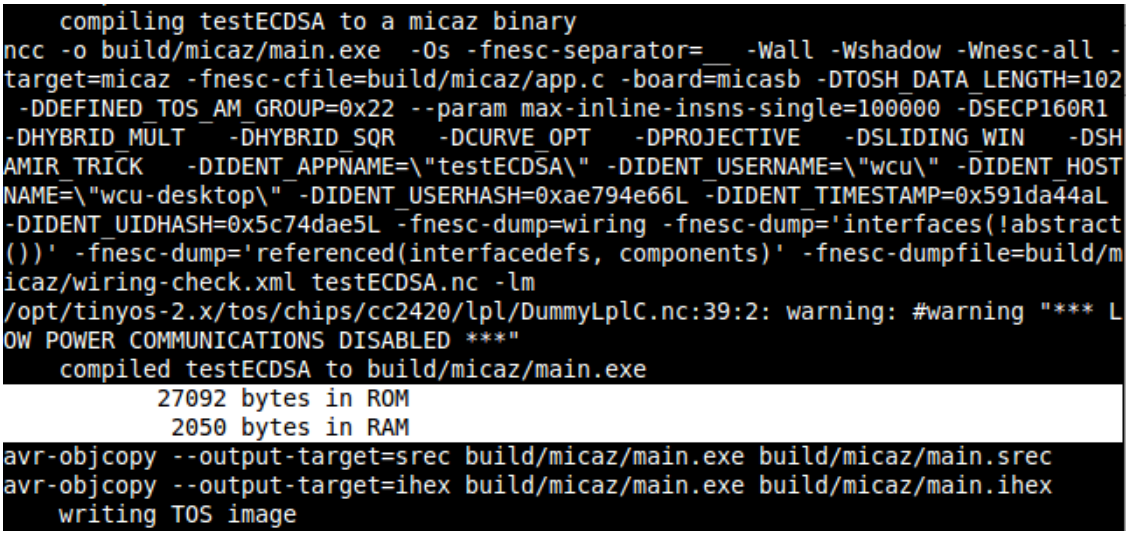

Figure 4.9 Evaluation of required memory in ECDSA

❖ **Energy consumption**

Figures 4.10 and 4.11 present the results of the energy consumption after the compilation of ECDSA for two phases:

- Signature phase

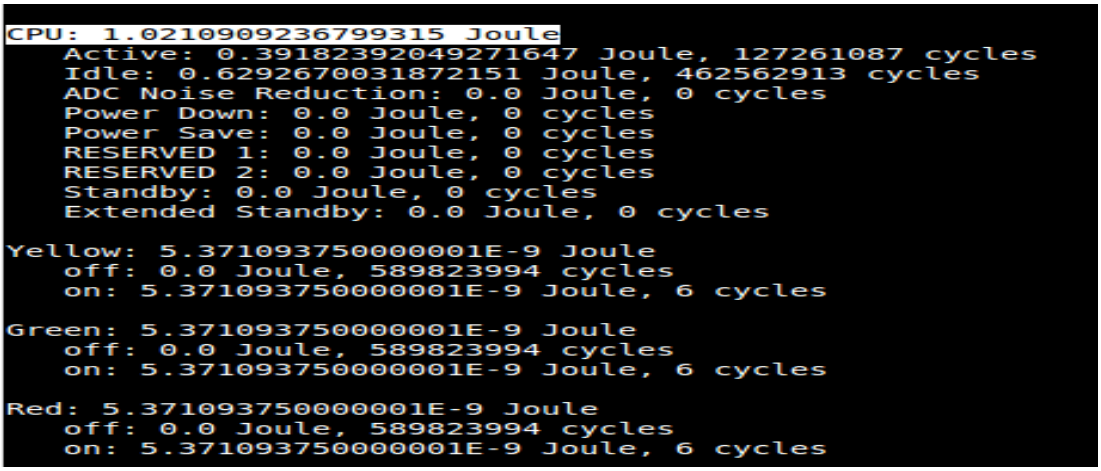

Figure 4.10 Evaluation of energy consumption in ECDSA

- Verification phase

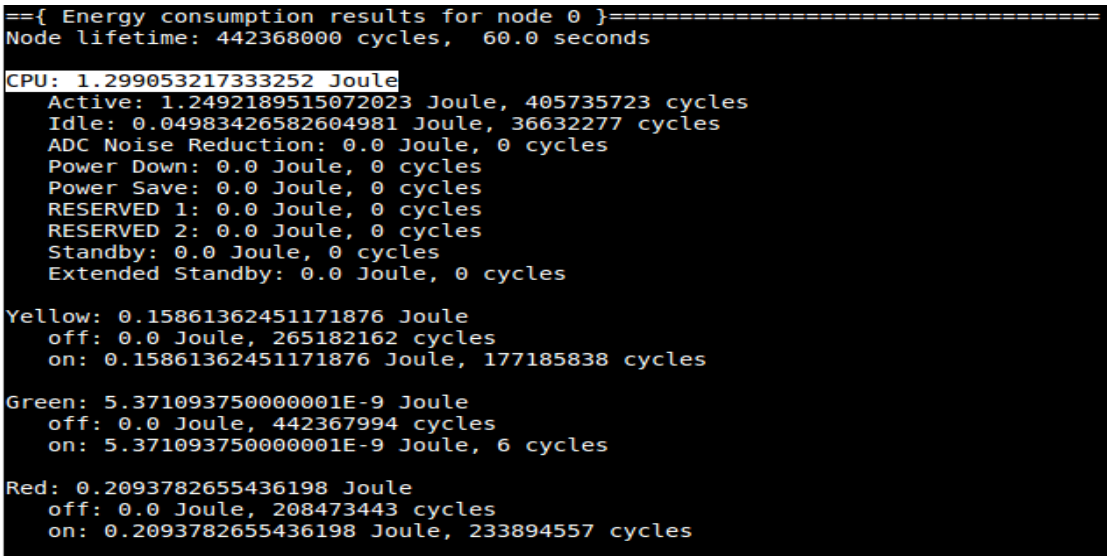

Figure 4.11 Evaluation of energy consumption in ECDSA

❖ **Execution time**

The figures below show the evaluation of the execution time in the ECDSA after the compilation with two phases:

- Signature

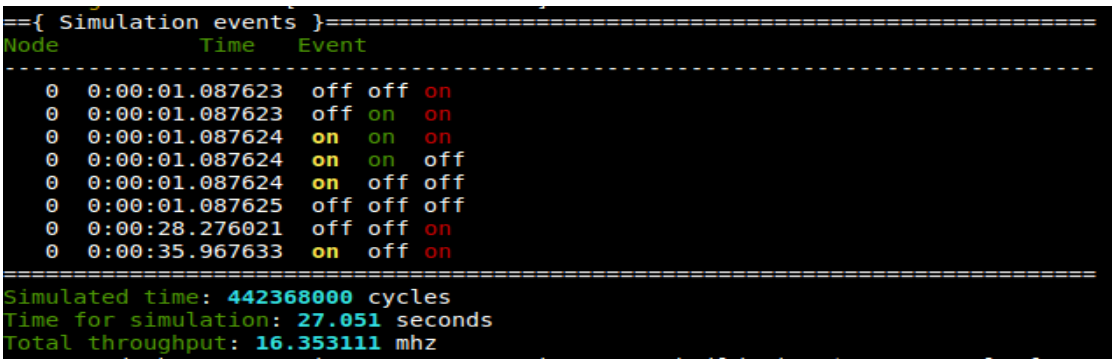

Figure 4.12 Evaluation of execution time in ECDSA

- Verification

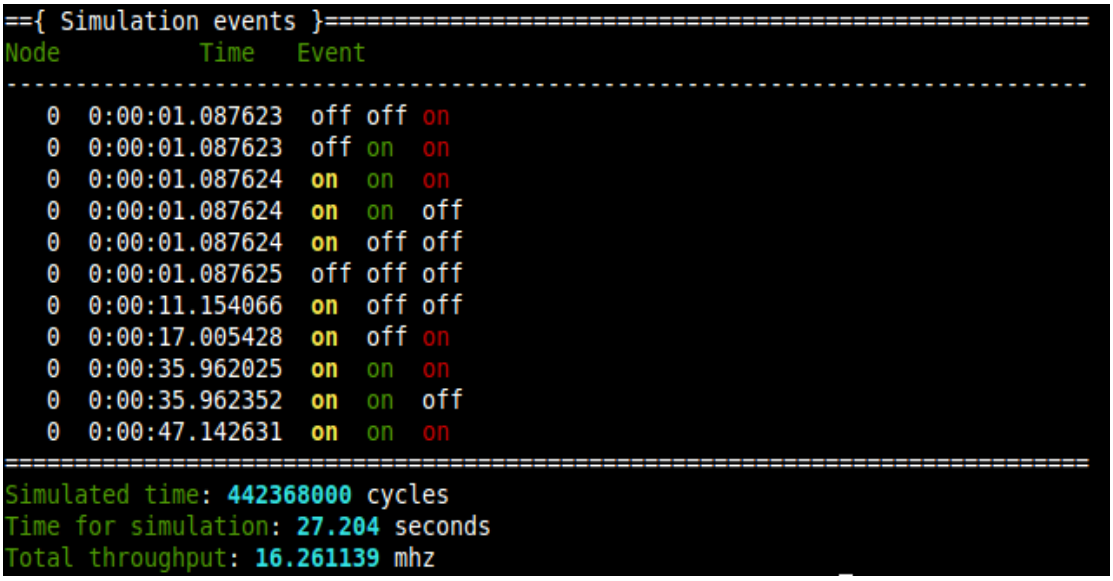

Figure 4.13 Evaluation of execution time in ECDSA

4.5.3 Moon Protocol

❖ **Required memory**

Figure 4.14 and figure 4.15 show the results obtained after the compilation of Moon protocol for two phases:

- Signature phase

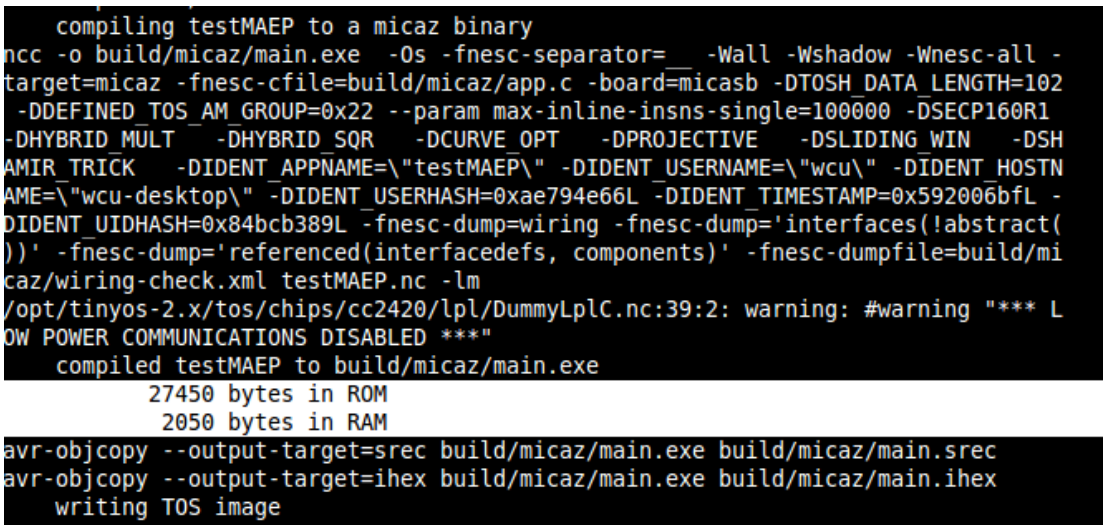

Figure 4.14 Evaluation of required memory in Moon protocol

- Verification phase

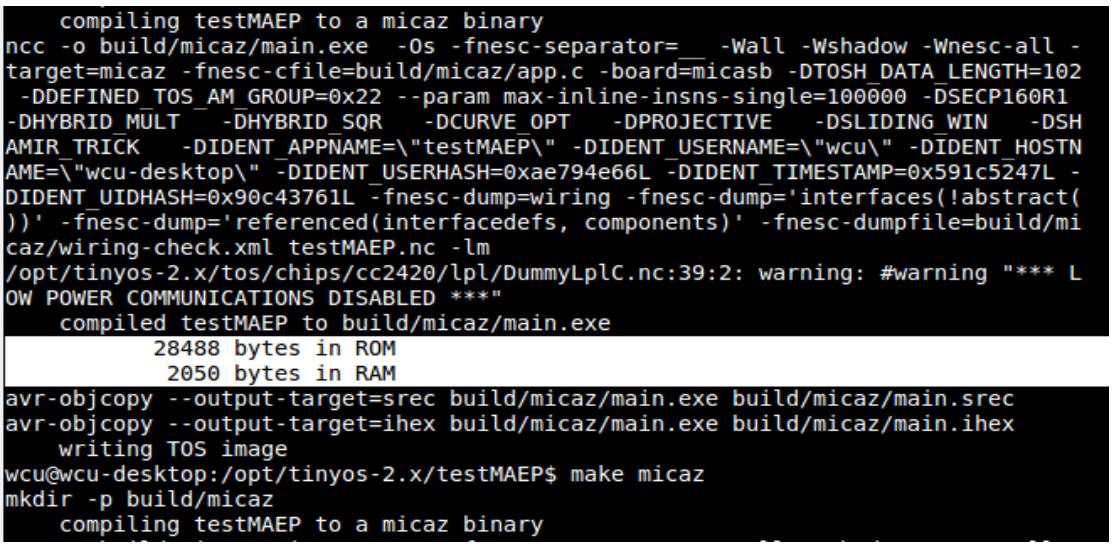

Figure 4.15 Evaluation of required memory in Moon protocol

❖ **Energy consumption**

we present in the next figures the results of the energy consumption after the compilation of Moon protocol for two phases:

- Signature phase

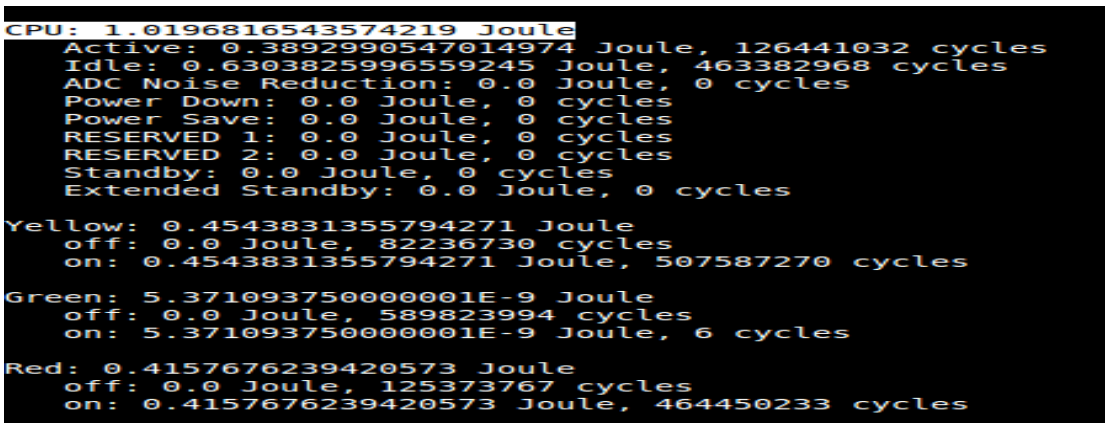

Figure 4.16 Evaluation of energy consumption

- Verification phase

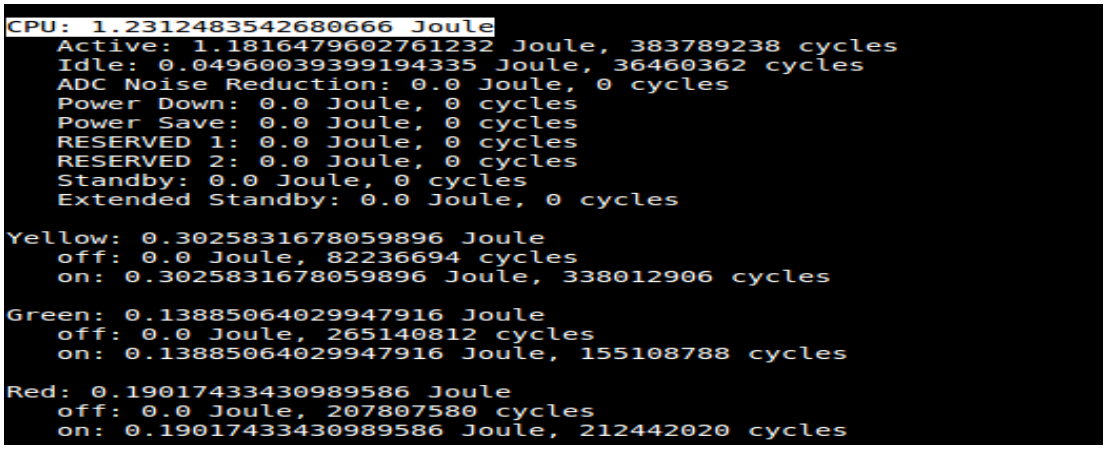

Figure 4.17 Evaluation of energy consumption

❖ Execution time

Execution time evaluation after the execution of Moon protocol as shown in the next figures

- Signature phase

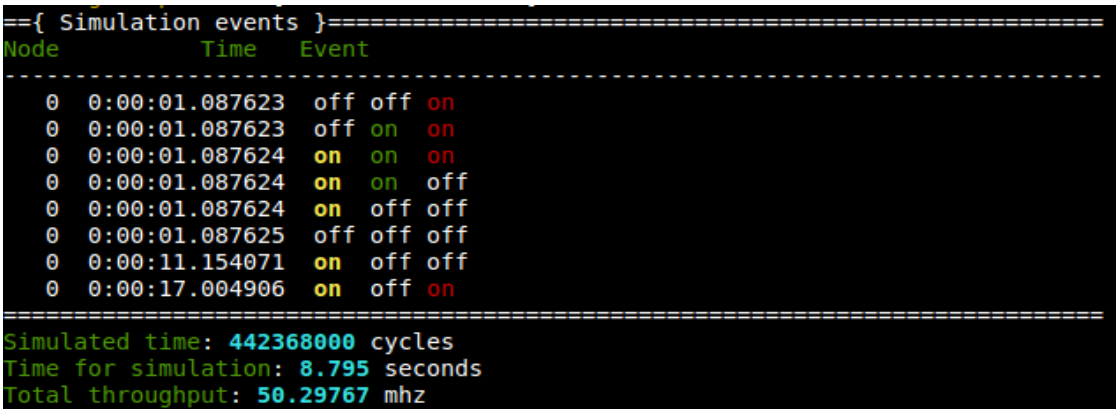

Figure 4.18 Evaluation of execution time in the Moon protocol

- Verification phase

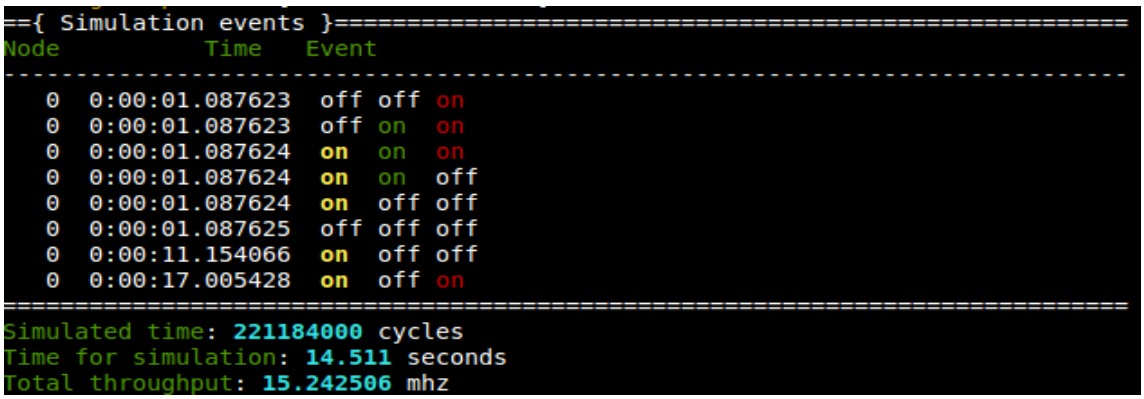

Figure 4.19 Evaluation of execution time in the Moon protocol

❖ **Results evaluation for ECDH**

Table 4.1 presents the obtained results of ECDH execution in term of performances.

	ROM	RAM	Energy consumption (Joule)	Execution Time (Second)
ECDH	**26352**	**2394**	**1.298728**	**7.031305**

Table 4.1 performance simulation for ECDH

❖ **Results evaluation for ECDSA and protocol of Moon**

Table 4.1 illustrate the obtained results of the performances simulation in ECDSA and protocol of Moon in term of signature and verification.

	Signature				Verification			
Performance	**ROM**	**RAM**	**Execution Time (Second)**	**Energy consumption (Joule)**	**ROM**	**RAM**	**Execution Time (Second)**	**Energy consumption (Joule)**
ECDSA	**25422**	**2050**	**7.691612**	**1.021090**	**27092**	**2050**	**11.180279**	**1.299053**
Protocol of Moon	**27450**	**2050**	**5.850835**	**1.019681**	**28488**	**2050**	**5.851362**	**1.231248**

Table 4.2 performance simulation for ECDSA and protocol of Moon

4.6 Results Discussion

In this section, we are going to discuss the obtained results from the previous tables after the simulation of the protocols evaluated in term of performance as follow:

ROM reservation

- According to the table 4.1 ECDH method required 26352 bytes.
- In the other side ECDSA required 25422 bytes and Moon protocol required 27450 bytes in term of signature phase as shown in table 4.2.
- For the verification phase we noticed that both methods, the ECDSA and the Moon Protocol required much more ROM than it required for the Signature phase that the ECDSA required 27092 bytes and Moon Protocol required 28488 bytes. As the obtained results indicates we found that the ECDSA is a better method when it comes to the required ROM used.

RAM reservation

- The value of RAM reservation in ECDH is 2394 byte (table 4.1).
- For the ECDSA and Moon protocol, they required the same value, which is 2050 byte in the both phases signature and verification.

Execution time

- The time processing value in the ECDH is 7.031305 second.
- For the signature phase, it took ECDSA 7.691612 second and Moon protocol 5.850835 seconds.
- Whereas in verification phase it took ECDSA 11.180279 and Moon protocol 5.851362 seconds. As result Moon protocol is better when it comes in term of execution time.

Energy consumption

- The results of the table 4.1 shows that the ECDH took 1.298728 Joule.
- The best result shown in the table 4.2 when comes to energy consumption is 1.019681 Joule for signature and 1.231248 Joule for verification phase which are obtained using the Moon protocol method. This results shows that the Moon protocol consumes less energy compared to ECDSA and ECDH.

For the EKE protocol that we have described in chapter 3, this protocol uses expential function which is requires an important energy consumption and a long execution time. The present results illustrated on the tables shows that the protocol of Moon is the most efficient in term of execution time and energy consumption compared to ECDSA. So Moon protocol currently is suitable security protocol for wireless sensor network.

4.7 Conclusion

In this chapter, we evaluated the protocols performance through simulation on RAM and ROM memory size, execution time, and energy consumption. these protocols which are implemented by the NesC language and simulated in TOSSIM and Avrora simulator in the TinyOS environment network. The results show that the most efficient protocol is the Moon protocol.

GENERAL CONCLUSION

WSN are innovative research subjects for various disciplines of Information and Communication Sciences and technics, but with specific constraints that pose challenges. Among the problems currently posed in this type of network is security. the sensor node limitation resources generate several design challenges, the most important of which are security and energy saving (resource security). So, the objective is to conceive protocols that offer a good level of security while respecting the metrics such as energy consumption, size memory and processing time.

The present work highlighted the essential characteristics of wireless sensor networks, as well as the security needs and challenges in wireless sensors. We have also studied some security protocols based mainly on asymmetric key that provide the basic security service for any communication-based system which are:

- Elliptic Curve Diffie Hellman (ECDH).
- Elliptic Curve Digital Signature Algorithms (ECDSA).
- Protocol of Moon et al that we have implemented by using the NesC language and the TinyOS environment.

We have used the TOSSIM and Avrora simulators to make a comparative study between these protocols in terms of RAM and ROM occupation, energy consumption and processing time of the selected sensor type which is Micaz.as a result we found that protocol of Moon is the most efficient and suitable for security in wireless sensor network.

As a prospect, we will continue to improve our work, and especially take into account other metrics as implementation of authentication protocols more resistant to different attacks with low energy consumption and short execution time.

BIBLIOGRAPHY

[1] A. Boudhir, Contributions to the energy optimization for security, localization and routing in wireless sensor networks, Doctorat, University Abdelmalek Saaidi ,2013.

[2] A. Hassen.Moon, RajaUmmer.Iqbal, Authentication Protocols for WSN using ECC and Hidden Generator, Anna University, India, Volume 133 – No.13, pp.4 ,2016.

[3] A. Hassen.Moon, RajaUmmer.Iqbal, G. Mohiuddin.Bhat**,** Authenticated key exchange protocol for Wireless Sensor Networks, University of Kashmir, India**,** Volume 11., pp.3 ,2016.

[4] A. Hassen.Moon, RajaUmmer.Iqbal, G. Mohiuddin.Bhat**,** Mutual Entity Authentication Protocol based on ECDSA for WSN, University of Kashmir, India, pp.6 ,2016.

[5] A. Mnif. An ID-based User Authentication Scheme for Wireless Sensor Networks using ECC, University of Sfax, Tunisie ,2012.

[6] Crossbow , Wireless sensor network product refence guide , 2007 , available on : http://www.investigacion.frc.utn.edu.ar/sensores/Equipamiento/Wireless/Crossbow_Wireless_2007_Catalo , checked in : 8/02/2017 in : 19:31.

[7] C. De Cannière, B. Preneel, "Trivium – A Stream Cipher Construction Inspired by Block Cipher Design Principles", Katholieke University Leuven Belgium, 2002.

[8] C. Lederer1, et al, Energy-Efficient Implementation of ECDH Key Exchange for Wireless Sensor Networks, University of Klagenfurt, Austria, pp. 6, 2010.

[9] C. Yacine, « Réseaux de Capteurs Sans Fils », support-SIT60, Vol. 103, pp. 14-17, 2008.

[10] D. Boubiche, Une approche Inter-Couches (cross-layer) pour la Sécurité dans les R.C.S.F, Doctorat, Université de Batna.

[11] D. SOW, Cryptographie à clés publiques et Protocoles d'échange de clés, Doctorat, Université Cheikh Anta DIOP de Dakar,2013.

[12] F. Zhang, R. Dojen, Coffey, Comparative performance and energy consumption analysis of different AES implementations on a Wireless Sensor Network Node, University of Limerick, Limerick, Ireland,2014.

[13] H. MEKIDICHE, RAIS, La géolocalisation de réseaux capteurs (algorithme DVHOP), Master 2, University of Abou Bakr Belkaid, Tlemcen, Algérie, 2012.

[14] H. Nunoo-Mensah, K. Osei Boating. James Dzisi Gadze, Comparative Analysis of Energy Usage of Hash Functions in Secured Wireless Sensor Networks, International Journal of Computer Applications, Volume 109 – No. 11, pp.2, 2015

[15] H. Zhong, R. Zhao, J. Cui, X. Jiang, J. Gao, An Improved ECDSA Schema for Wireless Sensor Network, Anhui University, China, Volume. 9, No. 2., pp.3 ,2016.

[16] I. Mallouli, Implémentation d'une bibliothèque de fonctions cryptographiques optimisées pour les réseaux de capteurs sans fil, Ingénieur, L'Ecole Nationale d'Ingénieurs de Sfax, 2011.

[17] M. Boudia, agrégation des données et sécurité des réseaux de capteurs sans fil, université of Tlecmcen,2014.

[18] M. Messai, Sécurité dans les Réseaux de Capteurs Sans-Fil, Magistère, Université Abderrahmane Mira de Bejaia, 2008.

[19] N. Bounegta, N. Aici, Approche Décentralisé pour la sécurité d'un Réseau de Capteurs Sans Fil (RCSF), Ingénieur, Université de Bechar, 2010

[20] N. Chikouche, A. Abdelmalek Ghehioueche, F. Mezrag, "Simulation of Encryption Schemes for Wireless Sensor Networks», Master, Mohamed Boudiaf University, M'sila, pp. 4, 2016.

[21] N. Laroussi, Secure Routing in Wireless Sensors Networks, National Engineer School of Sfax, Master ,2012.

[22] N. Lasla, la gestion des clés dans les réseaux de capteurs sans fils, Magistère, Institut Nationale de formation en Informatique Oued-Semar, Alger, 2009.

[23] P. Ning, TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks, University, Raleigh, pp.2.2009.

[24] R. Aslanoğlu, group key establishment protocols: pairing cryptography and verifiable secret sharing, Master, School of Engineering and Science Institute of Technology, İzmir ,2013.

[25] R. Haakegaard, J. Lang, The Elliptic Curve De-Hellman (ECDH), pp.2, 2015.

[26] R. Kacimi, Technique de conservation d'énergie pour les réseaux de capteurs sans fil, Doctorat, Université de Toulouse France, 2009.

[27] S. ATHMANI, Protocole de sécurité Pour les Réseaux de capteurs Sans Fil, Magistère, Université De Hadj Lakhdar - Batna, 2010.

[28] S. Bouguer, étude et simulation comparative entre les réseaux de capteurs sans fils traditionnels et les réseaux de capteurs véhiculaires, Ingénieur, Université Tlemcen, 2012

[29] S. Gustavo Quirino, R. Admilson. L. Ribeiro, E. David Moreno, Asymmetric Encryption in Wireless Sensor Networks, University of Federal de Sergipe, Brazil**.**

[30] S. Jaydip, Security in Wireless Sensor Networks, India, pp.2

[31] S. Raja Rajeswari, V. Seenivasagam, Comparative Study on Various Authentication Protocols in Wireless Sensor Networks, Anna University, India**,** pp.2-3 ,2016.

[32] S. Maarouf, S. Ouadah, Implémentation et évaluation des schémas de routage sur une plateforme réelle de réseaux de capteurs sans fil, Master, Université Tlemcen, 2014.

[33] S. M. Bellovin, M. Merritt, "Encrypted key exchange: Password based protocol secure against dictionary attacks," in IEEE Symposium on Research in Security and Privacy. IEEE Computer Society Press, 1992, pp. 72–84.

[34] V. Gayoso Martinez, L Hernandez Encinas, C. Sanchez Avila, A Survey of the Elliptic Curve Integrated Encryption Scheme, Volume 2, 2010.

[35] Y. Faye, algorithme d'authentification et de cryptage efficace pour les réseaux de capteurs sans fil, Université de comté, France, 2014.

[36] Y. SHOU, Cryptographie sur les courbes elliptiques et tolérance aux pannes dans les réseaux de capteurs, Doctorat, l'Université de Franche-Comté, 2014.

[37] Y. Younes, Minimisation d'énergie dans un réseau de capteurs, magister, Université Mouloud Mammeri de Tizi-Ouzou ,2012.

[38] W. Tahraoui, Wireless Sensor Network, 2009

[39] Encrypted Key Exchange EKE , http://cetiseer.ist.psu.edu/bellare00authenticated.html , checked in :10/04/2017 in 10/04/2017 , in: 15:23.

[40] Tinyos Documentation Wiki [on line], http://tinyos.stanford.edu/tinyos-wiki/index.php/TinyOS_Documentation_Wiki. Checked in: 19/02/2017 in: 17 :16

[41] NesC language definition ,Wikipedia , [online] http://en.wikipedia.org/wiki/NesC , checked in: 20/02/2017 in: 22:22.

[42] NesC language ,[online] http://www.cs.wayne.edu/~hzhang/ checked in : 20/02/2017 in :15:30.

[43] TinyECC official page [online], http://discovery.csc.ncsu.edu/software/TinyECC/. Checked in: 11/03/2017.

[44] TOSSIM simulator , [online] http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM checked in : 06/02/2017.in : 00:20.

[45] TOSSIM simulator [online] http://www1.cse.wustl.edu/~jain/cse567-11/index.html checked in : 06/02/2017 in : 20:17.

[46] "PowerTOSSIM-Z: Realistic Energy Model in TOSSIM for the MicaZ Mote". [online]. Available in: https://www.cs.tcd.ie/scarbajrs/powertossimz/ checked in: 2/03/2017 in: 14 :30.

[47] Avrora simulator [online],http://compilers.cs.ucla.edu/avrora/ checked in : 5/03/2017 in: 16:16.

ملخص

تعتبر شبكات الاستشعار اللاسلكية تكنولوجيا متوسعة في العديد من المجالات والتطبيقات المختلفة التي تتطلب اجراءات امنية فعالة وذلك لتفادي العديد من الهجمات التي تعرقل وتحجب السير الجيد لهذه الشبكات. كما نجد ان الموارد المحدودة لهذا النوع من الشبكات تشكل عقبة رئيسية خاصة في مجال الامن والطاقة المستهلكة لأجهزة الاستشعار.

من خلال هذا البحث قمنا بدراسة بعض البروتوكولات الامنية المصممة لاقتراح حل يلائم هذا النوع من الشبكات. وركزنا على برمجة بروتوكول مقترح حديثا من طرف موون واخرون. حيث اعتمدنا على محاكات هذه البروتوكولات لأجل تقييم اداء البروتوكولات من حيث الذاكرة المستعملة، وقت التنفيذ والطاقة المستهلكة باستعمال المحاكي TOSSIMو Avrora.

كلمات مفتاحية: شبكات الاستشعار اللاسلكية، الامن,TinyOS , NesC.

Abstract

Wireless sensor networks (WSN) represent an emergent technology in many fields and various types of applications that require high security in order to avoid several of attacks, which make them vulnerable and disrupt the proper functioning of the network. One of the weaknesses of wireless sensor networks is the resources limitation, which constrains security and energy efficiency. In this research work, we interested to study some security protocols in order to propose a suitable solution to this type of networks. For the evaluation of these protocols, we depending in the simulation in terms of: required memory, execution time and the energy consumption by using TOSSIM and Avrora simulator.

Keywords: wireless sensor network, security, TinyOS, NesC.

Résumé

Les réseaux de capteur sans fil RCSF représentent une technologie émergente dans de nombreux domaines et différents types d'applications exigeant une grande sécurité pour éviter plusieurs attaques qui les rend vulnérables et perturber le bon fonctionnement du réseau. Cependant, la limitation de ressources des nœuds capteurs constitue une contrainte importante, principalement en termes de sécurité et d'autonomie d'énergie. Durant ce travail, nous étudions certains protocoles de sécurité faisant à proposer une solution adaptable a ce type de réseau. On base sur l'implémentation d'un nouveau protocole proposée par Moon et al. Pour l'évaluation de ces protocoles en s'appuyant sur la simulation en terme de l'espace mémoire, le temps d'exécution et la consommation d'énergie utilisant les simulateurs TOSSIM et Avrora.

Mots clés : réseaux de capteur sans fil, sécurité, TinyOS, NesC.