People’s Democratic Republic of Algeria

Ministry of higher Education and Scientific Research

UNIVERSITY of Dr. TAHAR MOULAY SAIDA

FACULTY OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCES

Master’s Thesis

Specialization:R.I.S.R

Theme

THE SECURITY IN WIRELESS SENSOR NETWORKS

Realized by:

- **Sekkoum Taha**
- **Kadri Malek**

Supervised by:

- **Mrs Taleb Fadia**

Abstract

In this thesis, we study wireless sensor networks and their security, starting with the sensor and its basic components, passing through to its uses, types, and fields of use. After we got acquainted with the sensor, we pass to the sensor networks and see what they are made up and how they are deployed, down to how their elements communicate with each other and exchange information. We also mention some of the usual attacks that apply to this type of network and how to protect against it, passing through what is considered the most important thing is security, which is the encryption and its types used in such networks.

Résumé

Dans cette thèse, nous étudions les réseaux de capteurs sans fil et leur sécurité, en commençant par le capteur et ses composants de base, en passant par ses utilisations, ses types et ses champs d’utilisation. Après avoir fait connaissance avec le capteur, nous passons aux réseaux de capteurs et de voir ce qu’ils sont constitués et comment ils sont déployés, jusqu’à la façon dont leurs éléments communiquent les uns avec les autres et d’échanger des informations. Nous mentionnons également certaines des attaques habituelles qui s’appliquent à ce type de réseau et comment se protéger contre lui, en passant par ce qui est considéré comme la chose la plus importante est la sécurité, qui est le cryptage et ses types utilisés dans ces réseaux.

ملخص

في هذه الأطروحة، ندرس شبكات الاستشعار اللاسلكية وأمنها، بدءاً من المستشعر ومكوناته الأساسية، مروراً باستخداماته وأنواعه ومجالات استخدامه. بعد أن تعرفنا على جهاز الاستشعار ، نمر إلى شبكات الاستشعار ونرى ما هي مصنوعة وكيف يتم نشرها ، وصولا الى كيفية عناصرها التواصل مع بعضها البعض وتبادل المعلومات. ونذكر أيضاً بعض الهجمات المعتادة التي تنطبق على هذا النوع من الشبكات وكيفية الوقاية منها، مروراً بما يعتبر أهم شيء هو الأمن، وهو التشفير وأنواعه المستخدمة في مثل هذه الشبكات.

THANKS

First of all, we would like to thank our Almighty God for giving us the privilege and the chance to study and follow the path of science and knowledge.

We express our deep gratitude and sincere gratitude to our mentor Mrs Taleb Fadia for agreeing to lead this report, her understandings and her valuable remarks.

We would like to thank the jury members for agreeing to review our work.

We would also like to thank all the teachers in the computer science department, for the teaching they have arranged for us during our training cycle.

Thank you to all those who, in one way or another, have contributed to the realization of this work, and which we cannot quote individually.

Taha and Malek.

Dedication

With the help of Almighty God, we were able to carry out this modest work that we dedicate:

To the dearest beings who sacrificed their lives for our happiness, who have always been by our side, in joy and sadness, our parents whom we love very much, for their support throughout our life, may God protect them.

To our whole family.

To all the teachers who contributed to our learning.

To all our friends and comrades and the promotion of Master 2 RISR.

And to those who gave us a hand

And for all those who love science.

Taha and Malek

Content

LISTE OF TABLES

LISTE OF FIGUES

GENERAL INTRODUCTION

Networks, in general, are an important achievement that humanity has made, as anyone connected to any network operating independently can share his information, experiences, and experiences on this network, which in turn is connected to a group of other independent networks. The role of these networks has evolved even when discovering new lands and areas, and here come the sensor networks, where this type of networks plays a major factor in exploring and studying areas of unknown topography for humans to facilitate human exploitation, knowledge, and proximity to them by studying the natural phenomena that occur in them throughout the year. Or for a certain period, where these sensors transmit valuable information that is difficult to obtain, with the lowest possible costs and resources, and all of this is to be fully informed of the nature of these studied areas and to understand more.

CHAPTER I
GENERAL INFORMATION ON WIRELESS SENSOR

I.Abstract

We start with this section, which focuses only on the sensor, its history, and first use, and then to its physical parts and components.

The sensor is characterized by having an operating system that differs from the usual systems. We also talk about its types, its sensing patterns, and some examples of that. Then we will look at some of the common problems it faces.

I.Resume

The first use of sensors was in World War II, where it appeared under the name And who had a role in sensing submarines and ships at sea, more information can be found in the historical section of this chapter.

The sensor usually consists of a Sensory (sensing) unit, Communication unit, Information processing unit, and Energy unit.

External memory, Serial Adapter, GPS can be added

It works as an operating system called tiny OS, Its types are passive and active sensors, And its patterns to binary sensing model and power-law sensing model.

Despite the difference in type and style, it shares several problems it faces, including the first and most famous of which is energy, as the sensor is limited in energy, in addition to the corruption of one of its main parts could be both hardware and software….etc.

I.1.Introduction

Feeling things is what makes us realize what is around us and understand our environment and our world. All this thanks to the various sensory cells that are in our body, thanks to which we can observe what is around us, gather information from our surroundings, move and act according to that information. For example, you can feel cold or hot, thanks to the sensory cells in your skin, you can smell delicious food or a smokethanks to the sensory pickups in your nose, and you can listen to the music or someone who speaks, thanks to a complex sensory system in your ear. In the same way, we can speak about the two other senses.

Humans have transferred this feature to the world of technology, and now the sensory pickups are around us in every device present on your TV, on your phone, in your car, on your computer, on the roads, in factories stores, on land, and at the sea. Collecting information has become an obsession and challenge to obtain knowledge, especially in environments that are difficult to access or remain in, such as volcanoes, ocean floor, space, and other planets. This is what leads us to our topic, which is the wireless sensor. So, let’s make something that collects information and sends it to us. It seems easy, but it is not the case.

Building a technology that captures natural phenomena information (temperature, pressure, humidity, etc.), then processes it and sends it to us from an unknown environment, relying only on the This complex component is typically composed of: microcontroller (processing unit), radio transceiver and receiver, sensory picker, external memory (ram and rom), battery or electric power source and serial adapter.

Now, let's take a break and talk about the first wireless sensory pickup.

I.2 .History

It is the height of the Cold War. Three hundred meters beneath the surface of the stormy North Atlantic, a Soviet submarine steams past the Icelandic coast. The Soviet Captain looks to his crew. Everyone is holding their breath waiting to find out if they've crossedthe formidable NATO anti-submarine picket line that stretches from Iceland to mainland Europe.

After several tense minutes of silence, the crew relaxes- sonar can hear NATO patrol ships far away, but not a single one of them has changed course, they haven't been detected.

Ordering his men to hold bearing, the Captain plots a course a few hundred miles from the American coastline where his nuclear ballistic missile submarine will loiter undetected, ready to deliver a devastating surprise nuclear attack in the case of war.

The Russians are good submariners, but their subs lack sophistication, and unbeknownst to them a powerful American underwater weapon can detect them from clear across the Atlantic, zeroing in the US Navy's hunter-killer subs onto their location.

For decades Soviet nuclear attack submarines believe that they are prowling the oceans of the world undetected, completely unaware of the hidden killers always following their every move.

If nuclear war ever broke out, the Soviet ballistic missile submarine fleet would never get a chance to join the war, eliminated in minutes by the hidden assassins keyed on to their locations by an incredible piece of American technology: The Sound Surveillance System, or SOSUS.

In 1937 Leigh University scientist Maurice Ewing made a critical discovery that would catapult American sonar technology far ahead of its competitors- while doing seismic refraction experiments in water three miles dept in the North Atlantic, Ewing used explosive charges placed at different depths to generate sound waves.

As Ewing listened to the echoes of the explosions, he discovered that sound signals at very low frequencies could travel great distances with minimal loss, and he postulated that in certain conditions so-called “deep sound channels” could exist which would propagate an acoustic signal for hundreds or even thousands of miles.

Wasting no time, the Navy immediately authorized a slew of tests for developing these deep sound channels for military use.

The test was a huge success, and a system for helping locate and rescue downed pilots were immediately developed,Named SOFAR, for Sound Fixing and Ranging.

A huge success, but some minds in the US military slowly began to see an altogether different.

By the early 1950s, the US government believed that Soviet submarines posed the greatest threat to American security over any other Soviet weapon, and thus established Project Hartwell.

For six months the best and brightest minds of the American Navy and civilian scientists alike drew together to discuss how to counter the Soviet submarine threat, Long-range submarine detection was premier in the list of topics discussed during Project Hartwell, and a focus of its efforts.

Then physicist Frederic Hunt electrified the project heads with a stunning, and very convincing idea: why not use SOFAR to detect Soviet subs?

He showed Project Hartwell's leadership that higher frequency sounds made by Soviet subs could be easily detected at ranges of a few hundred miles, but frequencies below 500 Hz would easily penetrate through the various layers of the oceans to reach the deep sound channel from virtually any depth, and thus make detection of noisy Soviet subs possible at ranges of thousands of miles!

The US Navy immediately started several highly secret research programs to begin building underwater listening stations.

This budding secret surveillance network was classified with the acronym SOSUS, standing for Sound Surveillance System.

With the first SOSUS contact on a Soviet nuclear boat west of Norway established in 1962, SOSUS would go on to play a major role during the Cuban missile crisis, when it detected three Soviet submarines leaving Russian waters and heading for Cuba, In 1968 SOSUS made its first detections of Soviet Charlie and Victor class submarines, proving its worth even against upgraded Soviet designs (6).

I.3.Wireless sensor

A sensor node is a small physical device, with low cost and low power, this component is equipped with on-board microprocessors, a specific operating system (1), and Information capture unit (The sensor unit), an Information processing unit, a transceiver unit and finally, a very limited energy unit (3).

The sensor lifetime is also limited: when the battery drops below a minimum charge level, the sensor stops working (2). That is why the integration of an application on this type of component must always take into account certain constraints: energy consumption, memory space (4).

It can also contain additional units such as the location system (GPS) to know the precise location of the node.

The nodes are deployed in hostile environments called the capture field, where it is very difficult, if not impossible, to change or recharge the batteries (1).

A sensor has the task of collecting data, transforms an observed quantity into a quantity usablein an autonomous way, data processing, and wireless communication with other sensor nodes or directly to an external base- station, with the aim of accomplishing a common task (1).

I.4.Architecture of a sensor node

I.4.1.Sensory (sensing) unit

It contains two subunits, the first one allows the collection of observed physical phenomena such as (temperature, pressure, humidity, etc.), the second one converts the signal to make it understandable by the treatment unit.

I.4.2.Communication unit "transceiver unit"

It is composed of a transmitter/receiver (radio module) allowing communication between the different nodes of the network. Note that the transmission consumes a lot of energy compared to the calculation unit (5).

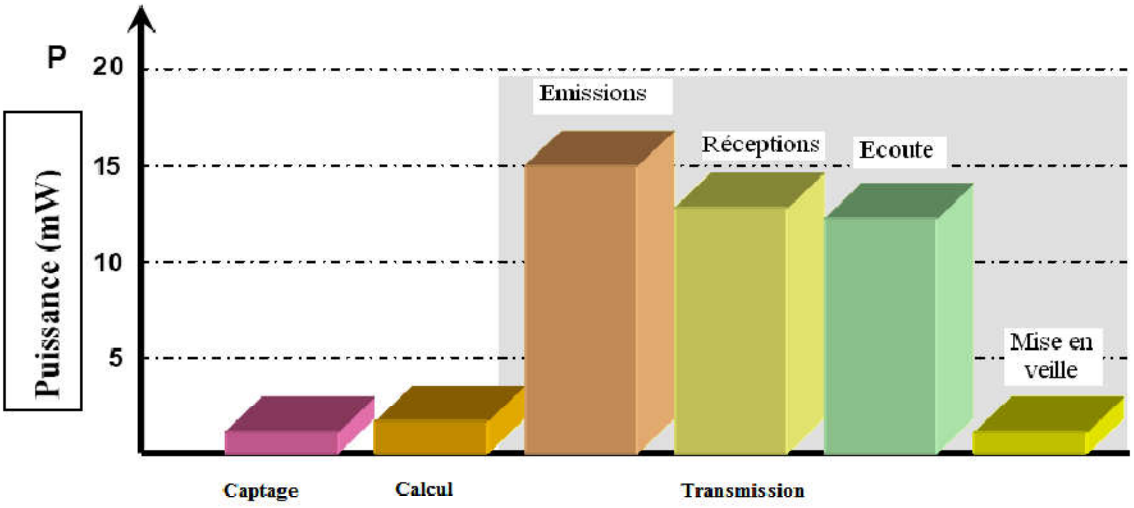


Figure 1 Energy consumption in the capture, calculation, and transmission (7).

I.4.3.Information processing unit:

This unit constitutes the central element of the sensor. It is composed of a processor and a memory integrating an operating system specially designed for sensor nodes. The best known operating system is called Tiny OS. This unit has two interfaces, an interface for the Sensory unit and an interface for the communication unit. It acquires the information from the Sensory unit and sends it to a communication unit. This unit is also responsible for executing the communication protocols which allow the current node to collaborate with other sensors nodes. It can also analyze the collected data (8).

I.4.4.Energy unit:

It provides the necessary energy for the operation of the device. However, the onboard devices being small, the batteries are also small and therefore the energy resources are limited. This limitation makes energy management a critical point for the sensor node and thus for sensor networks. But, the size of the batteries makes these systems more and more mobile. Therefore, their transport is easier. Additional components can be added depending on the field of application, for example, a localization system such as a GPS (Global Positioning System), an energy generator such as solar cells, or a mobilizer allowing it to move (9).

I.4.5.External memory

Very often, you can also find some external memory either onboard or an optional slot such as a MicroSD. This allows the possibility to permanently store some larger amounts of data instead of always streaming it to a sink/base station (10).

I.4.6.Serial Adapter

Almost, every sensor node comes with some sort of serial communication, usually a cable which can connect the sensor node to some other device (most often to your PC). The serial adapter is used for programming and debugging the node and sometimes for charging it (7).

I.5.TinyOS Overview

TinyOS is an "operating system" designed for low-power wireless embedded systems. Fundamentally, it is a work scheduler and a collection of drivers for microcontrollers and other ICs commonly used in wireless embedded platforms. TinyOS is written in nesC, a dialect of C language(15).

I.6.Types of Sensors

There are endless types of sensors available. Some are readily available to be used in particular platforms, whereas others need to be interfaced first to be used in your sensor node platforms. Some are very complex and expensive, whereas, others are considered as standards and are not very expensive. Some are large and others are tiny. Figure.2 offers some examples. Furthermore, you must differentiate between passive and active sensors: (11)

- Passive sensors:
Passively sense their environment without manipulating it. Very often they also do not need any power to sense their environment. Power is only needed to amplify the analog signal. Examples of passive sensors are thermometers, light sensors, and microphones(11).

-Active sensors:
Need to actively manipulate their environment to sense it, for example, by emitting light or sound waves. Active sensors need the power to sense. Active sensors examples are sonar distance sensors or some types of seismic sensors, which generate shock waves.

There is also a differentiation between Omni-directional and narrow-band or directional sensors. With omnidirectional sensors, the sensed phenomenon does not really have a position or direction, e.g., the temperature is sensed omnidirectionally. On the contrary, narrow-band sensors can sense the environment only in one direction or a small angle. For example, sonar distance sensors emit a sound wave in a given direction to calculate the distance to the next obstacle.
An example of a passive, but the directional sensor is a camera(11).

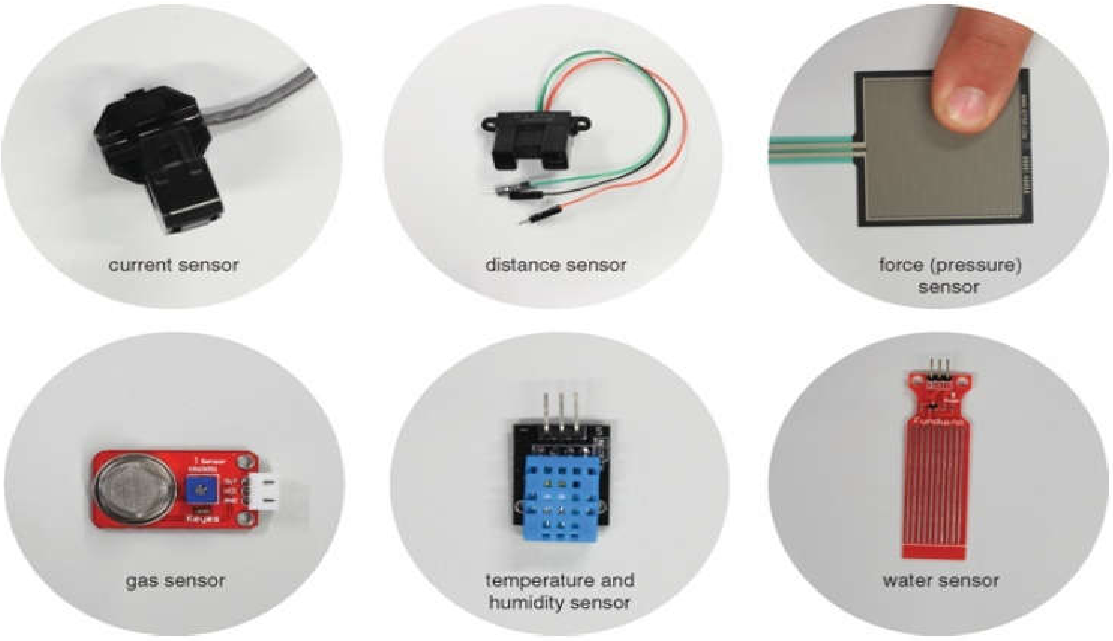


FIGURE. 2 Examples of various sensors(11).

I.7.SENSING COVERAGE

How far can a sensor sense its phenomenon? For example, if you have a movement sensor, what is the maximum distance of the movement that the sensor can actually detect? And what happens if this distance is exceeded? These answers are provided by the sensing coverage and sensing model. The coverage is the maximum distance at which the sensor still works reliably and correctly. The sensing model gives you information on what happens if this distance is exceeded(12).

-The binary sensing model:assumes that the sensor delivers perfect results inside its coverage area and faulty or missing results outside(12).

-The power-law sensing model:assumes that the sensor's reliability decreases with increasing distance from the phenomenon.

Binary and power-law are the two main sensing models. A combination of both is of course also possible. In practice, it is preferable to use the binary model based on real observations. For example, experiments can show what the maximum distance is from a moving object so that the movement sensor can detect it. This maximum distance is then considered your coverage area and you rely fully on it(12).

I.8.High-level Sensor

A high-level sensor is a sensor that does not directly correspond to any sensor hardware. For example, if the sensor is: movement in a camera image, you cannot simply ask the camera (at least typical cameras) to tell you whether there is a movement in its image or not. Instead, at least several individual images from the camera need to be processed, a decision can then be made about whether there was some movement or not. Another term often used for high-level sensors is sensor fusion. The term sensor fusion indicates that several sensor inputs need to be fused or combined together to have a new sensor.

For example, a high-level sensor could be called “house intrusion” and its low-level sensors could be “door code entered,” “movement sensor,” “door open sensor,” and “window open sensor.” All of these four sensors are fused together to obtain the “house intrusion sensor.”(13)

1.9.Node Problems

Node death is one of the most common problems. Suddenly, a node stops working completely. This might seem like a serious problem and it is. However, there are some positive points. When a sensor node suddenly stops working, its internal state remains valid. This means that whatever the data stored on the node, it will correctly remain there (unless the sensor does not store data at all). Also, neighbor nodes are usually well prepared for dead nodes and quickly exclude them from routing or co-processing. Thus, real node death is a problem, but a manageable one. However, what sometimes looks like a dead node is often not a completely dead one. This can be due to variousreasonssuch as:(14)

- **Slowly deteriorating batteries:** causing individual components on the node to perform badly or not at all.

- **Faulty components:** causing errors while sensing, storing, or communicating, while the general behavior of the node is not affected. Faulty components couldbeboth hardware and software.

- **Node reboots:** typically take time and cannot be easily detected by neighbors. They are often caused by software bugs, during this time; the node seems to be unresponsive and uncooperative.

- **Link/Path Problems:** the biggest issue is interference. Interference can occur between individual nodes of the network but also between the nodes and external devices such as mobile phones, Bluetooth devices, or a high-voltage power grid. You cannot see the interference and it changes quickly and unpredictably. In fact, interference by itself causes “only” lost packets. However, these sporadically lost packets trigger an avalanche of other problems at all levels of the communication stack. They cause more traffic because packets need to be resent.

- **Global Problems:** these problems affect the network's general work and can be coarsely classified into topology, lifetime, and semantic problems.

- **Topology problems:** Missing short links are links between very close neighbors that are not able to communicate with each other. These links are expected and planned by the designer to offer alternative communication paths. The opposite problem, unexpected long links, occur when two neighbors with an unexpected long distance between them are able to communicate well. Both problems cause premature battery depletion of some nodes. Another difficult topology problem caused by missing links is **partitioned networks.** In these networks, nodes in one partition are completely isolated from the nodes in other partitions. This causes the partition to spend a lot of energy to search a path to the receiver and never send its data.

- **Lifetime problems:** refer to an unexpectedly short network lifetime. The time when useful data is gathered. This time might start, immediately after deployment or sometime after it. It ends then the network is not able to retrieve as much data as it needs to semantically process it. Obviously, this definition will have different interpretations in different applications. Lifetime problems occur most often when individual nodes use their energy too quickly and deteriorate the network's general performance. The most endangered nodes are the ones close to the sink or at communication knots(they serve as forwarders for many other nodes). However, software bugs also can cause them to stay awake for extended periods of time and waste their energy.

- **Semantic problems:** occur most often when the data and its meaning do not match Semantic problems expectations. This can be caused by missing or bad calibration of the sensing hardware; by deteriorating batteries causing wrong sensor readings or by a wrong interpretation of the data itself.

I.10.conclusion

The sensor nodes are small units with cheap components in order to make millions of them because we need a lot of them. We can consider them as a huge army led by one leader who is the sink node. There are several types and patterns as needed, each type works differently from the other and shares many problems together.

CHAPTER II

WIRELESS SENSOR NETWORKS

II.Abstract

In this chapter we take a general look at WSN,it's the definition and different types of WSN than to its main architecture,moving on to the protocol architecture that is developed for this kind of networkand of course it's characteristics and then how to deploy this sensor.

II.Resume

Wireless sensor network (WSN) is an AD HOC network. A collective term to specify an independent setof tiny computers with the main target of sensing some physical property of their environments, these kind ofnetworks are different depending on the use so there are types for houses and factories,there are kinds for medical use and there are the very last ones that focus on making every object smart Enoughto serve the human.

WSN has it's own OSI model that consists only of 5 layers witch are:The application layer,Routing,Link management,Medium access (MAC)layer.

WSN is a limited network with limited resources that uses economical protocols and cryptography to help the networkto function longer,faster,and provide good results with minimum resources and no infrastructure.

The hardest kind of these networks is those who deployed on a hostile area by throwing theme from a plane witch is the Randomly scattered deploymentor placing the network nodes one by one witch is Deterministic deployment.

II. Introduction

In the previous chapter, we talked about sensory pickups and their role in our daily life. Now we went to the next question, which is: how these sensor nodes and cells communicate with each other and exchange information? Perhaps, there must be a mediator, isn't it?

The answer is: of course. There is always a link between two things that want to communicate.from fire signals in the past to homing pigeons to message bottles in the sea.

In sign language, there are always protocols and rules to communicate and give information to the other party. For example, if someone raised a white flag, it is known that the white flag indicates surrender. When your friend stands far away and makes movements, you notice that with your own eyes. All these thanks to the light. The photons reflect on his hand and spread in the air, and when they reach your eyes, they are light waves. Similarly, about hearing, credit goes back to the sound waves.

So, what about machines? How can they communicate? what technologies are used and what rules should be taken? We will discuss all this in this chapter.

II.1.Definition and presentation of sensor networks

Wireless sensor network (WSN) is an AD HOC network. A collective term to specify an independent set of tiny computers with the main target of sensing some physical property of their environments: such as vibration, humidity, or temperature. They consist of a few to thousands of sensor nodes, which are connected via wireless communications. Typically, there is also at least one special node, called the sink or the base station, which connects the sensor network to the outside world(16).

Example: Forest fire detection.

The wireless links are spontaneous and not planned. Different from other wireless networks, such as Wi-Fi hotspots, WSNs are not carefully planned to perfectly communicate and enable specific service quality levels. Instead, the assumption is that each of them tries to detect its brothers and sisters and to exchange some minimally required data with them. Thus, WSNs are deployed (installed) quickly and without much knowledge of the environment(17).

II.2.Wireless sensor networks

II.2.1.Types of Wireless sensor networks

A. **Cyber-physical system (CPS):**This is a newer term for a wireless sensor network. It attempts to better describe what you can do with these networks and their main properties when being integrated into a physical environment. Different from other computers and devices, which are environment agnostic. Cyber-physical systems are part of the environment and application restricted. Another important property is the fact that they also can affect the environment via so-called actuators, such as in automatic irrigation pumps, light switches, alarms, and humidity or temperature regulators. Interestingly, the term wireless is no longer used in cyber-physical systems. The reason being, many sensor networks are now wired, since power supply in buildings is easier to maintain through wired connections than through wireless ones. Smart homes are a typical example of cyber-physical systems(18).

B. Body sensor networks: Refer to a specific type of network, designed to be carried on the body (mostly human). Applications include health monitoring, weight management, sports logging, and many others. There are some peculiar examples, such as smart shoes or smart T-shirts, which can sense your activity or heart rate. Most of the sensor nodes are tiny, sometimes even implantable(18).

C. Participatory sensing, collaborative sensing, or crowdsourcing: This refers to a new and fast-developing type of sense in which sensors are essentially humans with their smartphones. For example, people can track their biking paths and then evaluate them in terms of safety, noise, or road quality. All these data are gathered on a central point and processed into a single biking quality map of a city, which can be distributed to any interested user. The real power of these applications is that no additional hardware is needed, only a simple user-oriented application for smartphones. At the same time, this is also the largest challenge in these applications, motivating people to join the sensing process and to deliver high-quality data. Privacy is a big issue with these applications because it requires people to share their location data(18).

D. **The Internet of Things (IoT):** Is often mistaken to be a sensor network. However, IoT's main concept is that all things, such as a washing machine or radio, are connected to the Internet. The Internet connection has significant advantages when implementing sensor networks and can thus be seen as an enabling technology. However, the target can also be very different, e.g., when you can read your emails from the microwave or from your car(18).

II.3.Hardware architecture

Usually, wireless sensor networks are built around the four main following entities:

II.3.1.The sensor

As its name says, it is in charge of measuring a relative value in the environment (temperature, pressure, brightness, presence, etc.). We can sometimes meet sensor-actuators which will not only measure but will also have for the role of taking an action according to the measured value. Intelligence is necessary for decision-making. The action to be taken can then be deported on another node of the network(19).

II.3.2.The aggregator

It is responsible for aggregating the messages received from several sensors and sending them in a single message to the sink. The main operation aims to limit traffic on the network and therefore extend the overall life of the network sensor(19).

II.3.3.The well

It is the end node of the network. It is to him that all of the values measured by the network are communicated. There can be several wells on the same network of sensors(19).

II.3.4.The gateway

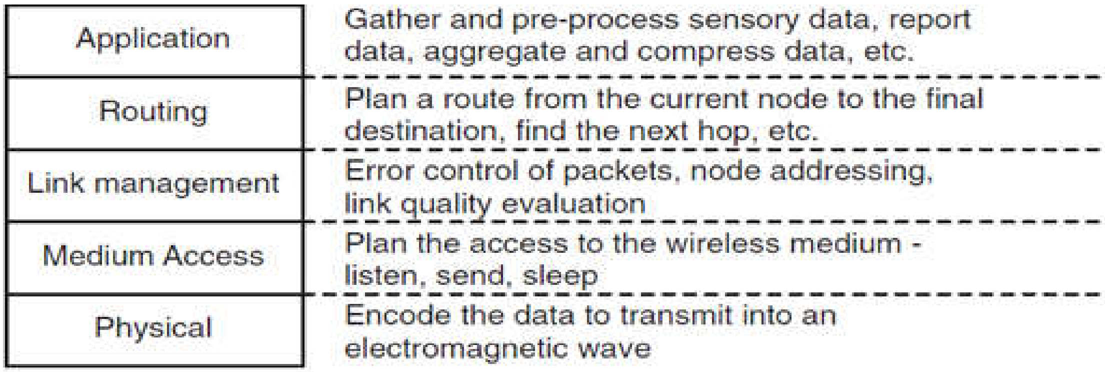
It is a device that has the distinction of having two interfaces network. It allows connecting the network of wireless sensors to a more traditional network, typically the internet(19).

II.4.Protocol architecture

II.4.1.Communication stack

The software of a sensor node is typically organized into smaller pieces called modules. Instead of implementing everything into one mixed application, often called spaghetti code, programmers organize their code into modules, which can be easily (or at least more easily) reused with other applications. The communication stack is such an organization, where different tasks of the inter-node communication are organized into pieces.

The most widely used communication stack is the OSI model, which underlies the Internet. It consists of seven layers. The lowest one is the physical layer, where physical communication takes place (e.g., electromagnetic waves for wireless communications). The highest layer is the application such as an email client or a browser. In between, there are five other layers handling things such as who is talking when and how to route packets from distant nodes(20).



Application	Gather and pre-process sensory data, report data, aggregate and compress data, etc.
Routing	Plan a route from the current node to the final destination, find the next hop, etc.
Link management	Error control of packets, node addressing, link quality evaluation
Medium Access	Plan the access to the wireless medium - listen, send, sleep
Physical	Encode the data to transmit into an electromagnetic wave

Figure 3. The simplified OSI model as typically used for sensor network applications(20).

II.4.2.Sensor Network Communication Stack

The OSI model is slightly too complex for sensor networks. Thus, we use the so-called simplified OSI model presented in Figure.1It consists of the following layers, which are connected(20).

A. **The application layer:** Is the top-most layer and is the actual application of the sensor node. For example, it can regularly sense the temperature and humidity of an environment, pre-process the data (e.g., compare it to some thresholds), compress the data, and send it to the network’s sink. Note that at this communication level, no information exists about where the sink is or where the node itself is(20).

B. **Routing:** Is the layer that plans communication in a broader scope. It knows how to reach the sink; even if it is several hops away (every node used to forward a packet to the sink is known as one hop). The routing layer abstracts the network as a communication graph, where paths between individual nodes exist(20).

C. **Link management:** covers the details about node-to-node communication. For example, it evaluates the communication quality between individual nodes; it corrects errors resulting from communication, and it takes care of giving addresses to nodes(20).

D. **Medium access (MAC):** Is the layer where the node manages its access to the wireless medium. Since the wireless medium (the air) is a shared medium, it is important to decide exactly when to listen, send, sleep(20).

E. **Physical layer:** Is the layer where the bits and bytes of individual packets get transformed into an electromagnetic wave that is physically transferred from one node to another(20).

II.5.Characteristics of sensor networks

It is completely different from other networks. The sensor network is wireless and generally used to communicate with the surrounding environment through a sensor. It means that it connects the environment to the machine, not a machine to a machine as usual.

If we talk in general, in some of these networks there is interaction with the environment (fire extinguishing systems).

If we talk between the network and the environment, this type transmits information in one direction only (from the environment to the network), but in the networks that we know, several machines exchange information between them all of the time.

These networks are Networks without infrastructure and decentralized architecture.

II.6.Deployment

II.6.1-Randomly scattered:

A contract can be randomly deployed from an aircraft; random deployment is supported in most scenarios for many reasons, such as cost and time.
Random deployment cannot provide uniform distribution over the area of interest, which leads to new problems in sensor networks. The main problems are location, area coverage, communication, and security(21).

II.6.2-Deterministic deployment (Placed at a specific location):

The nodes can be placed one by one in a deterministic way by a human or a robot. Deployment can be done all at once or it can be an ongoing process by redeploying other sensors in the same area. In many applications, manual deployment is impossible(21).

II.7.Wireless Access (Interference)

Sometimes, your home WIFI network decreases, you may experience slow internet connection, low signal strength, or a slow transfer when trying to share files between 2 computers using WIFI. But why this is happening?(22)

What is Wireless Interference?

Wireless interference occurs when something interferes or weakens the wifi signal. Wireless networks use radio waves for transmitting the data, but without a good wireless transmission, it is hard to obtain a reliable signal. So, what causes the down performance of the wireless network?(22)

II.7.1.RANGE

If the connected device is too far away from the signal source, the signal strength would be low or could be gone, as the wireless network has a limited range.(23)

II.7.2.DEVICES

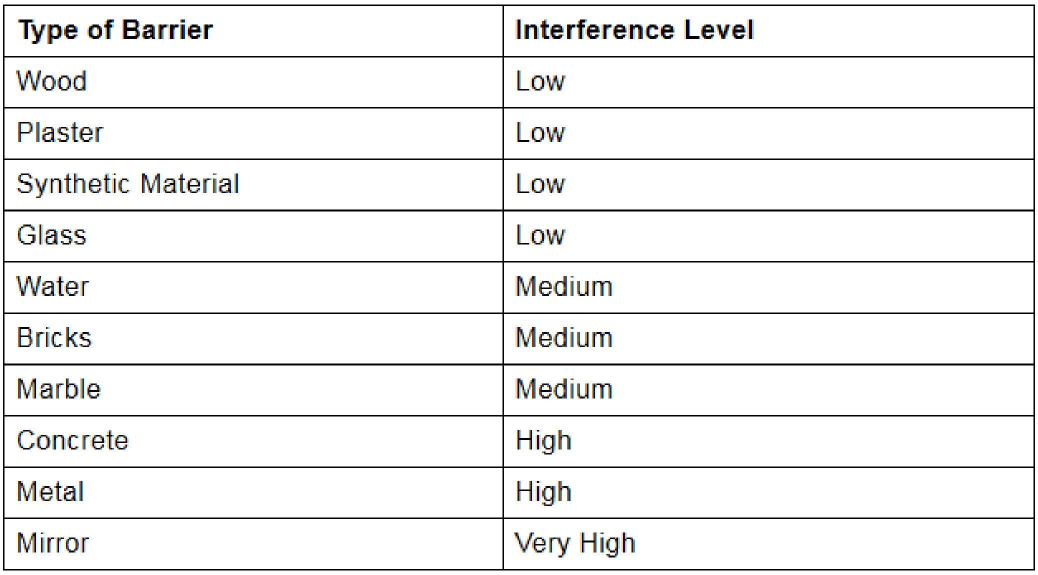
Nearly, any electronic device that produces electromagnetic signals can cause network interference. Common devices such as microwave oven, wireless video camera, wireless game controller, Bluetooth devices can cause interference because they use the 2.4Ghz frequency band, the same frequency band used by your router.(23)

II.7.3.bandwidth and channels

Bandwidth is the amount of data transmitted. The amount of bandwidth is divided between all devices connected to the same access point. So, if too much bandwidth is used, such as downloading a large file or watching a high-quality video, all the parts of the network will slow down.(24)(25)

II.7.4.Barriers

“Physical barriers, such as the materials that a building is constructed with, is one of the most likely reasons for a WiFi signal being unable to reach your device”(26) common building materials are:



Type of Barrier	Interference Level
Wood	Low
Plaster	Low
Synthetic Material	Low
Glass	Low
Water	Medium
Bricks	Medium
Marble	Medium
Concrete	High
Metal	High
Mirror	Very High

Table 1. Common building materials and their level of interference.

II.8.Network Topology

Network topology is the layout of a network. It defines the communication between nodes or devices within a network. There are two categories of topologies wired and wireless.(27)

II.8.1.Wired topologies:

II.8.1.1.Types of wired topologies

A. Star Topology

All devices are connected to a central computer, these devices communicate across the network by passing data throw the central computer.
Advantage: If one pc goes down it won't affect the other pcs.
Disadvantage: If the central computer goes down, all pcs are effected causing what's known as a single point of failure (SPOF) and the entire network is down.

B. Ring Topology

Devices are connected in the form of a closed-loop or a ring.so each device as two neighbors for communication purposes. The data packet is sent around the ring until it reaches its final destination.
Advantage: Easy to install.
Disadvantage: If one pc goes down or there was a break in a cable, the data flow would be disrupted.

C. Bus topology

Each device is connected to a signal coaxial cable or backbone using a special connector called BNC connector and the cable is terminated using terminators.

Advantage: cheap and easy to implement.

Disadvantage: If a pc or one of the terminators is removed from the network, the cable would be opened and that will cause a signal reflection and the data would be disrupted.

D. Meshtopology

Each device is connected to every other device within the network.this is commonly know as the internet.

Advantage: If one or more connections failed, the devices would be still able to communicate with each other.

Disadvantage:Expensive.

E. Hybridtopology

It is the combination of two or more topologies:

Advantage: Very helpful for a large network.

Disadvantage: Costly.

II.8.1.2.Types of wireless topologies

A. Infrastructure topology:

This topology is combined with wired and wireless topology. For example, as shown in figure 4 below we have 3 devices connected to a switch with a wireless access point connected to this switch by a cable ., this is similar to a star topology. The wireless access point can connect other devices to the network wirelessly and it acts as a bridge between the wired and wireless network.

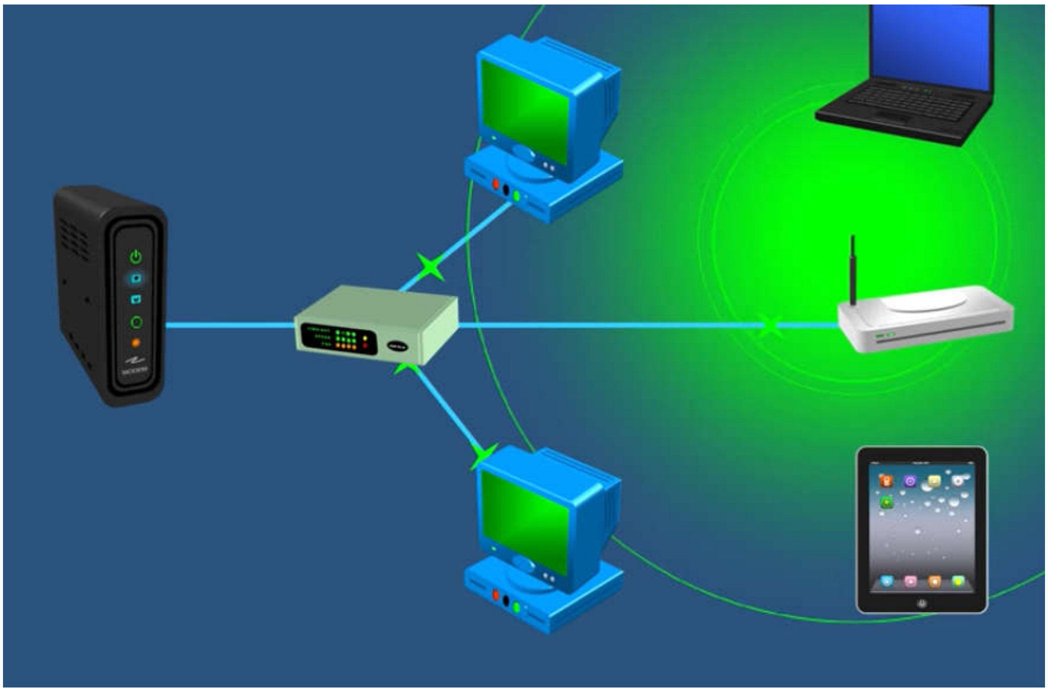


Figure 4.Infrastructuretopologyexample.

B. Ad hoc topology:

Ad hoc network does not require any infrastructure such as routers, web servers, cables.All devices wirelessly connect to other devices in a peer to peer network.they are directly connected with the lack of a central device such as a router or wap and because of that, each device is responsible for its security and permissions.

C. Wireless meshtopology :

Similar to mesh topology, only devices are connected wirelessly. For example figure 5, in a building, we have a WAP(wireless access point) connected to a switch and the switch connected to a modem all via cable. The other WAPs existing in the building can connect to the modem by routing the way to the WAP that's connected to wirelessly.

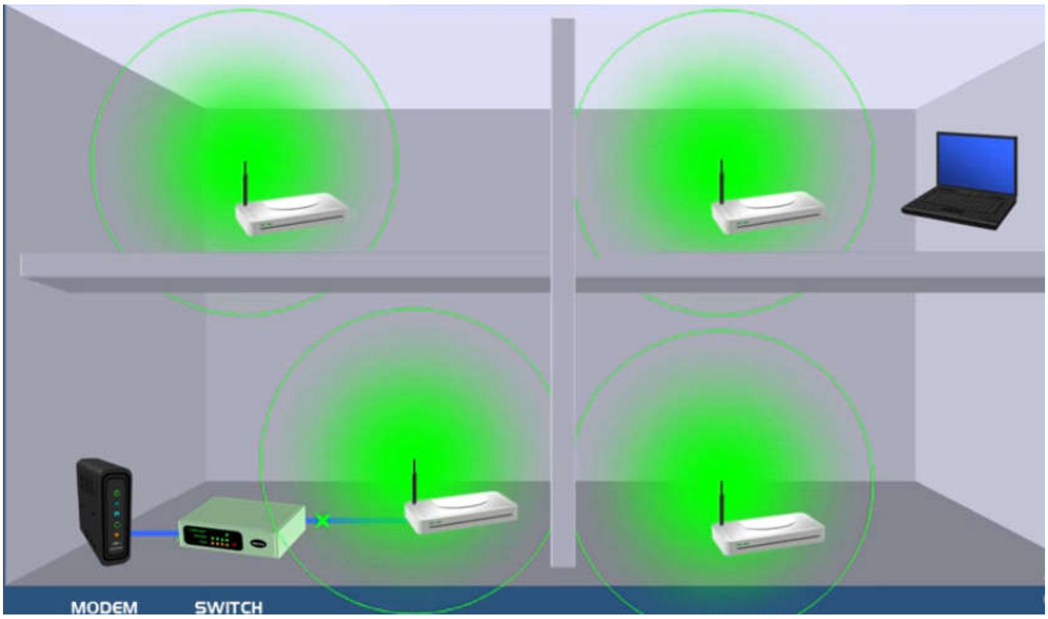


Figure 5.Wireless mesh topology example**.**

II.9.Limited Resources

Wireless sensor network devices are low in terms of battery, memory, and CPU. Since they use limited resources, they operate as long as they can. A longer operation can cause a big problem for the system which requires maintenance such as applying a resource adaptation policy.(28)

II.9.Limited bandwidth

Bandwidth is the rate at which data is transferred to a device within a network, And it's measured by the amount of data that is transmitted through the physics medium such as cable or wifi, generally measured in (bits/s).

Bandwidth is the range of frequencies needed to pass the signal in which is Loaded with data. For example, your ISP (internet service provider) gives you bandwidth and by that, you can tell how fast you cand downloaded or upload data from the internet. If you have a high bandwidth you can easily download or upload things. However, if you have a low-level bandwidth you'll find it very hard and frustrating.

Bandwidth limitations can cause a slow data transfer or even a data loss.there is a limit on how much data can be sent especially when using a wireless network when the target is too far away, or when exposing to interference such as a noise like poor quality cabling.(29)(30)(31)

II.10.Areas of application of the WSN

The Wireless sensor network covers a wide range of areas like Military, Environmental, Home, Commercial, Area monitoring, Health care monitoring, Earthsensings, Air pollution monitoring, Forest fire detection, Landslidedetection, Water quality monitoring, and industrial monitoring.(32)

II.10.1.Military Applications

Wireless sensor networks play a big role in the military security application. By using its functionalities, the military can monitor the enemy movements and coordinate the army activities. For example, as shown in figure 6 there are motes deployed around the area, these motes are connected with sensors allowing it to monitor the area and send periodical messages to headquarters. In case of any suspicious movements, the motes will immediately send a message to HQ which will take the necessary actions to deal with this situation like informing the command in charge of that area.(33)

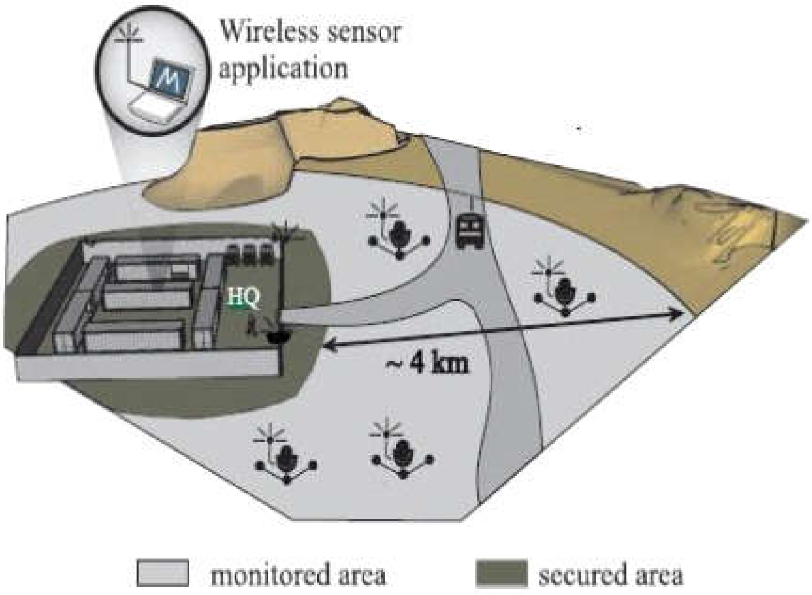


Figure 6.Military Applications example**.**

II.10.2.Environmental Applications

Environmental monitoring is one of the most compelling industrial applications driving transparency and efficiency across a variety of industries such as manufacturing, mining, oil and gas, and agriculture.

Monitoring and managing environmental conditions such as air quality water quality and atmospheric hazards is critical to prevent adverse conditions that may impact production process product quality equipment and worker safety.

The Environment Monitoring System Applications has seen great development in recent years in agricultural monitoring, habitat monitoring, indoor monitoring, greenhouse monitoring, climate monitoring, and forest monitoring to help people in their works and reduce the risk, cost, and time.(34)(35)

II.10.3.Medical Applications

Wireless healthcare monitoring has been applied by inventing Wireless Body Area Networks, a new generation of Wireless Sensor Networks.

This technology allows us to sense the human body's biological information and transmit it wirelessly over a short distance using wearable and implantable sensors. These sensors deliver acquired information to either a control device on the body or placed in a reachable location. And then, the data is assembled in the control device and sent « to remote destinations in a wireless body area network for diagnostic and therapeutic purposes by incorporating other wireless networks for long-range transmissions ».(36)

II.10.4.Precision Agriculture

Precision Agriculture or (PA) is a technology for the agricultural system. The purpose of this technology is to improve the agricultural processes by monitoring each step to give a better production with less impact on the environment. With the help of wireless sensor networks, precision agriculture started emerging.(37)

« PA requires a unique software model for each geographical area, the intrinsic soil type, and the particular crop or plants. For example, each location will receive its optimum amount of water, fertilizer, and pesticide ».(38)

II.10.5.Home Applications

Home AUTOMATION (application) is also known as an intelligent building is becoming more popular these days. This system provides household activity such as integrated control of lighting, heating, ventilation, and air conditioning. It also provides security such safety locks of gates and doors and so on...the system uses a lot of energy to do its tasks.so to reduce the unnecessary energy consumption, the

WSN technology is used. The main purpose of it is to monitor and control the parameters like voltage, current and temperature.this way we can increase the performance of the smart home.(39)(40)

II.11.Wireless sensor networks routing protocols.

II.11.1.Routing protocol

A routing protocol is an operation that takes or finds the best path for the data to arrive at its destination. This operation faces some difficulties such as finding the best suitable route, type of network…etc.(41)

II.11.2.Challenge

Designing a routing protocol for WSNis quite challenging for example:

1. Allocating identifiers, that means that sensor nodes don’t work very well when using internet protocol
2. Sensor nodes use batteries which means that the energy capacity is limited. Routing protocols should be designed in the best situation of energy consumption in the network.

3. Sensor nodes are small and so limited when processing data.

4. *Scalability,* in a network, while deploying thousands of sensor nodes, the routing protocol must be designed to handle this great number of nodes perfectly.

5. Delay: the routing protocol must provide a minimum delay for transmitting data because some applications require instant response without any possible delay that may cause due to a sensor warring.

II.11.3.Classification of routing protocols

The communication between nodes and how the data or the information is traveled throw the network, all of this is defined using routing protocols. We can classify the routing protocols for WSN in many ways such as the classification shown in figure 7.(41)

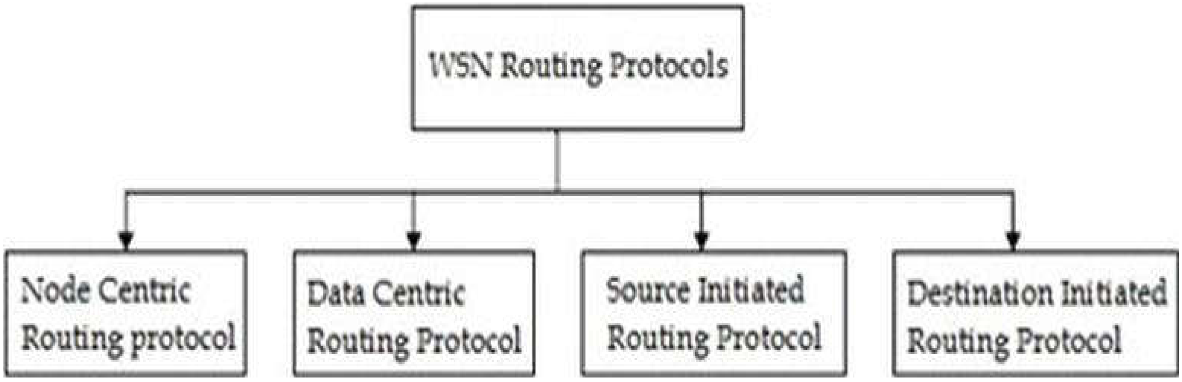

WSN Routing Protocols
Node Centric Routing protocol
Data Centric Routing Protocol
Source Initiated Routing Protocol
Destination Initiated Routing Protocol

Figure 7.classification of routing protocols.

II.11.3.1.Node centric

In node centric protocol, the destination node is determined with a special numeric identifier (or numerical addresses).The data can be sent by the node to the desired destination using the node's ID.However, this is not suitable for wireless sensor networks such as LEACH or Low energy adaptive clustering hierarchy.(41)

II.11.3.1.1.Low energy adaptive clustering hierarchy

Low energy adaptive clustering hierarchy know as (LEACH) orders the cluster in WSN. Clusters are constructed of sensor nodes and one of them is defined to be the cluster leader or the cluster head. This node will be the routing node for the rest of the nodes in the cluster.

The cluster leader or the cluster head is selected before the communication begins. If the communication fails somehow that means that there is a problem with the cluster head. Randomly a new cluster head is selected from the node group and this selection is announced to the rest of the nodes.(41)

II.11.3.2. Data-centric

Data-centric plays a big role in transmitting sensed data by a specific attribute instead of collecting data from different nodes in the network.The data-centric routing technique routes the sink node queries to a special location for collecting the data.(41)

II.11.3.3.Source-initiated

In this protocol,a source node will alert if it has any shareable data and then it will generate a data route to the destination.(41)

II.11.3.4.Destination-initiated

This type of protocol is called when the data route is generated from the destination.

II.11.4.Routing Protocols categories

Flooding and gossiping protocols are the two main techniques used for transmitting data in sensor networks. With the flooding protocol, once the data is received by the sensor nodes, the data is broadcasted to other neighbors. The broadcasting process will not stop until the data packet is being transmitted successfully to its proper destination, and the packet hops have reached its maximum number. As for gossiping protocol, when the sensor node gets the data packet, it transmits it to a selected neighbor.After that, the sensor node selects another node randomly and sends data to it, and this process keeps on going. (41)

II.11.4.1.Route discovery

II.11.4.1.1.Reactive protocols

These kinds of protocols operate only when a sensor node sends data to another node, which means that the routes get created on a request with the queries initiation.Some of the reactive routing protocols are :

II.11.4.1.1.1.Ad-hoc on-demand distance vector routing system (AODV)

Ad-hoc on-demand distance vector routing system (AODV) on-demand algorithm, a routing protocol designed for mobile networks. It creates routes between nodes only when needed by the source node, and it chooses the shortest free path from the routing table for transmitting packets.(42)

II.11.4.1.1.2.Dynamic source routing (DSR)

Dynamic source routing (DSR) developed at CMU 1996, this routing protocol can be on-demand or reactive. With this protocol, every source can define the route for transmitting packets to the desired destination. This protocol contains 2 parts :

Route discovery: defines the path for transmitting packets between source and destination.

Route maintenance: makes sure that the path is optimum without any loops.(43)

II.11.4.1.2Proactive protocols

With these protocols, routes are set up before using even using them and they keep the routing table for the entire network by sending network information from node to node. A most common one is:

II.11.4.1.2.1.Optimized link state routing (OLSR)

Optimized link state routing (OLSR) a proactive routing protocol designed for mobile networks.It is based on the MultiPoint Relays (MPRs) and It provides routes for each node. « Using MultiPoint Relays reduces the size of the control message »(44)(45)

II.11.4.1.3.Hybrid routing protocols

Hybrid routing protocols have the advantages of both reactive and proactive routing protocols.

II.11.4.2.Network organization based routing protocols

II.11.4.2.1.Flat topology

For the flat topology, all nodes have the same attributes, which means that all nodes are the same. For example :

- Gradient-based routing (GBR)
- Cougar
- Constrainedanisotropic diffusion routing (CADR)
- Rumor routing (RR).(41)

II.11.4.2.2.Hierarchical based routing

In some networks, the Hierarchical based routing protocols are used when there are some nodes that they are stronger than other nodes, But not always. The best-case scenario for the uses for these protocols is when grouping the nodes for constructing a cluster, It is easier and uses less energy. For example:

- Threshold sensitive energy-efficient sensor network (TEEN)
- Adaptive threshold sensitive energy-efficient sensor network (APTEEN)
- Low energy adaptive clustering hierarchy (LEACH)
- The power-efficient gathering in sensor information systems (PEGASIS)
- Virtual grid architecture routing (VGA)
- Self-organizing protocol (SOP)
- Geographic adaptive fidelity (GAF).(41)

II.11.4.2.3.Location-based routing (geo-centric)

Here the nodes use localization protocols that allow them to identify their locations. For example :

- SPEED
- Geographical and energy-aware routing (GEAR)
- SPAN(41)

II.11.4.3.Operationbasedroutingprotocols

Routing protocols are classified to :

- Multipath routing protocols
- Query-based routing
- Negotiation based routing
- QoS-based routing
- Coherent routing(41)

II.11.4.3.1.Multi-path routing protocol

Multi-path routing protocol creates more routes (Multiple paths) for the data to arrive at the destination. This technique provides a better network performance with low delay, and if there was some kind of failure, the Multi-path routing protocol creates another route (alternative path) for the data.Some protocols are :

- Multipath and Multi SPEED (MMSPEED)
- Sensor protocols for information via negotiation (SPIN)(41)

II.11.4.3.2. Query-based routing protocol

Here the nodes will send data only if the destination node asks for it. The destination node sends a query demanding some data information and when the specific node senses the information it transmits it back to that node. For example :

- Sensor protocols for information via negotiation (SPIN)
- Directed diffusion (DD)
- COUGAR(41)

II.11.4.3.3.Negotiationbasedroutingprotocols

We keep the unnecessary data transmission at a minimum level using Negotiations. The data transmission will be made after the negotiation when the sensor nodes negotiate with other nodes. Theyalsoshare information about resourceavailability. Examples :

- Sensor protocols for information via negotiation (SPAN)
- Sequential assignment routing (SAR)
- Directed diffusion (DD)(41)

II.11.4.3.4. QoS basedroutingprotocols

« A routing mechanism under which paths for flows are determined based on some knowledge of resource availability in the network »(46).Examples are :

- Sequential assignment routing (SAR)
- SPEED

- Multipath and Multi SPEED (MMSPEED)(41)

II.11.4.3.5. Coherent data processing routing protocol

The nodes operate at the minimum in Coherent data processing routing protocol on the data before sending it to the other nodes. And then data is collected from different nodes and passed to the sink node.(41)

II.Conclusion

Wireless sensor networks are a technology that led to great developments in a wide range of applications. Sensors are used to monitor different types of environmental and physical conditions.however wsn devices have a limited resource so it's better to use a resource-saving method if the system comes across a critical situation.

CHAPTER III
WSN ATTACKS AND SECURITY

III.Abstract

This section focuses on the various attacks that the network can be exposed to, and on some procedures and protocols used against these attacks, which are classified according to the source according to the layer that the attack targets, starting from the physical layer to the application layer where we explain the mechanism of the attack and the resulting results, then we explain How to defend or protect against this type of attack.

III.Resume

Security in wireless sensor networks is centeredonsix fundamental requirements, namely: authentication, confidentiality, integrity, reliability, availability, and data freshness.

Because of the properties f the WSN these properties can lead to various types of attacks.

We discuss theme depending on the targeted layer,starting with physical the layer that can be attacked by Tempering,Jamming,Eavesdropping, and Traffic Analysis,that we can defend against with spread spectrum communication and encryption/cryptography.

Moving to the second layer and that’s the link-layer, this layer can be attacked by MAC Protocol Violations,MAC Identity Spoofing that we can defend against with Error-correcting codes Incorporating,rate-limiting in the MAC protocol, and the use of time-division multiplexing to limit the amount of time that nodes can use the channel,MAC identity spoofing can be tackled by associating each identity with a secret key.

Moving on to the layer that has a lot of attacks witch is Network Layer it can be attacked by these attacks: Sinkhole attack, Wormhole attack, HELLO Flood attack,acknowledgment Spoofing, Selective Forwarding, Routing Table Overflows, we can defend against these attacks using multiple paths for data transfer, protocols based on end to end authentication like SEAD and Ariadne protocols,precise clock synchronization between communicating nodes can help overcome attacks such as the HELLO floods or acknowledgment spoofing.

Moving on to the next layer,Transport Layerthat can be attacked using Denial of Service and Connection Desynchronization and we can defend against theme by using:the authentication of all packets being exchanged during a connection and connection by solving a puzzle so we don’t waste our resources.

And lastly, the Application Layer that can be attacked using Data Aggregation Distortion,Clock Desynchronization, Selective Message Forwarding that we can defend against using:Multipath routing,outlier detection algorithm,use authentication methods.

Then we discuss the three types of cryptography which are symmetric, asymmetric, and hybrid, and give some examples and simple explanation of some cryptography protocols, such as AES, DES, ECC……etc.

III. Introduction

Some people may ask, as I did in the beginning about why we are already protecting a network that is far from us, and sometimes it is not even possible to reach a place where it is difficult because of the environment and the answer lies in that at all, because it is far away from us, so it must be protected. Just false information coming from these networks may cost a costly reaction Or unnecessary and sometimes it may cost a person's life to make an excuse, for example, disrupting a network of heat pickers in a forest. This means that in the event of a fire, we cannot know this early and we will react quickly and let us take for example sending false information from the network and therefore be acted based on information False and wasting various resources based on this false information.

Now I see why we should protect this type of network.

III.1.Security in WSNs

Security in wireless sensor networks (WSNs) is centered on at least the following six fundamental requirements, namely: authentication, confidentiality, integrity, reliability, availability, and data freshness.

Why WSNs are Predisposed to Attacks?

In this section, we describe unique properties that predispose WSNs to various types of attacks.

III.1.1.Resource Limitations

Many security mechanisms employed in computer networks rely on some form of cryptography. While public-key cryptography is in many cases a more versatile security scheme relative to private key cryptography, the limited memory, and processing power of the WSN nodes means that the former method can only be used subject to a very careful optimization of the algorithms at both the design and implementation levels. This challenge does not arise with traditional computer networks(47).

III.1.2.Large-scale Deployment

Sensor networks typically contain very many nodes that work in collaboration to achieve the purpose of the network. This means that the two-party functionality seen with many security protocols may not always be suited for WSNs. Further, the large number ofnodes sharing sensitive information requires that security designs have to be cognizant of the fact that a single compromised node could leak sensitive informationabout the entire network(48).

III.1.3.Open Deployment

Because sensors are typically deployed in outdoor unattended environments, an attacker could very easily have physical access to them and extract sensitive information from the motes. This is in contrastto traditional computer networks where it is for the most part safe to assume threat models in which an adversary will have a very low likelihood of physically accessing a node(48)..

III.1.4.Wireless Connectivity

Because WSNs communicate via wireless communication channels, the adversary only needs to tune into the frequencies being used for communication to be able to eavesdrop on traffic being exchanged between thenodes. Different from a wired computer network, therefore, security mechanisms for WSNs need to take these kinds of attacks into consideration.

The unique attributes such as those described above make possible a number of attacks against WSNs that call for defenses, which are for the most part specificallytailored to WSNs(48)..

III.2.Security Requirements

The fundamental security requirements of a WSN are given below.

III.2.1. Authentication

Authentication enables a node to confirm that the identity of another node with which it communicates is as claimed. This helps a node to verify the origin of packets sent to it, ruling out the chance that spoofed packets or packets maliciously injected into the wireless communication channel may be mistaken for genuine packets. A message authentication code (MAC) attached to a message canbe used to verify the origin of a message(48)..

III.2.2.Confidentiality

Confidentiality stipulates that information is only accessed by nodes that are supposed to access it. Confidentiality is ensured by encrypting the packets being sent out at the originating node so that they are decrypted at the receiving node. Depending on the application, encryption could be applied to the data part of a packet or the full packet (including the header). Encryption of the full packet helps obfuscate the node identities (which are located in the header) which helps minimize the chances that a given node's identity could be spoofed by an eavesdropper(48)..

III.2.3.Integrity

Integrity ensures that a message sent between two nodes is not modified by an adversary. If for instance routing or clock synchronization packets exchanged between nodes are modified by a malicious entity, the entire network could come to a halt. A keyed checksum (e.g., a MAC) can be used to determine whether messages have been modified. Figure 8 illustrates how a MAC appended to a message at the sending node can be used by the receiver to determine the integrity of the message. Taking the message and a key as input, the sender uses a MAC algorithm to compute the MAC, which is then appended to the message before it is sent out to the receiving node. At the receiving node, the message and the key (same as the key which was used at the sending node) are input to the MAC algorithm so as tocompute a MAC. If this MAC matches with the one retrieved from the message, then the message is confirmed not to have been tampered with(49).

III.2.4.Availability

Availability indicates that the WSN must be functional at all times and provide services whenever needed(49).

III.2.5.Data Freshness

WSN systems typically either continuously sense and forward data from the environment or forward data in response to a certain event. In both cases, it is crucial that data sensed by the nodes reach the base station as soon as possible (i.e., when is still fresh). This does not only minimize the likelihood that replayed packets sent by an attacker could be mistaken for legitimate packets but also ensures that the right kinds of interventions can be undertaken to react to the events sensed by the network. In a target tracking application, for instance, the target can only be tracked if current data is forwarded to the base station(49).

III.2.6.Self-organization:

A wireless sensor network is typically an ad hoc network, which requires each sensor node to be independent and flexible enough to self-organize. There is no fixed infrastructure available for network management in a sensor network. Self-organization brings a great challenge to the security of the wireless sensor network(50).

III.2.7.The localisation:

Often, the utility of a sensor network will be based on its ability to automatically locate every sensor in the network. A sensor network designed to detect anomalies will need precise location information in order to pinpoint the exact location of a fault(50).

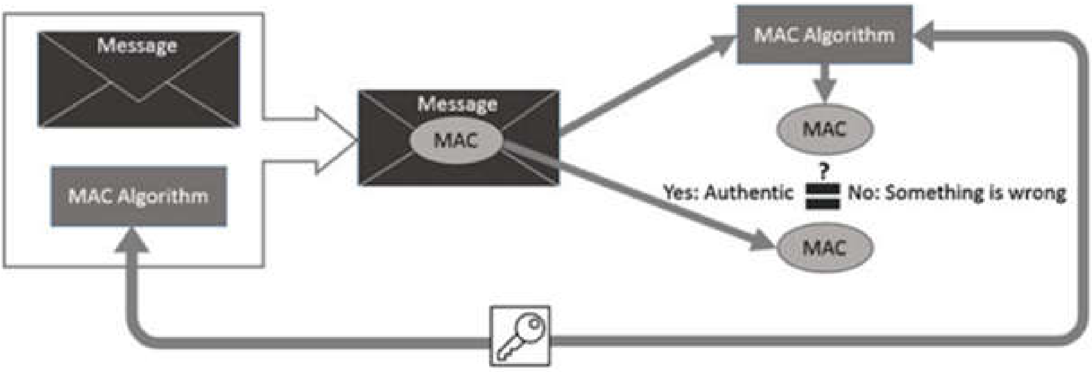


Figure 8.Using a MAC to verify message integrity(49)**.**

III.3.WSN Attacks and Defenses

There are several approaches to the categorization of attacks on WSNs. One of these approaches categorizes attacks based on whether they disrupt the functionality of the network or not. Attacks that disrupt network functionality are called active attacks (e.g., network jamming attacks, while those which do not disrupt network functionality are referred to as passive attacks (e.g., packet eavesdropping attacks. Another common way to categorize WSN attacks is to classify them based on whether they are internal (i.e., launched by nodes which are part of a WSN) or external (i.e., launched by nodes or devices that are not part of the network).

It is noteworthy that an attack launched by an external entity that gets authorization to access the network and then exploits the privileges to launch attacks would be classified as an internal attack.

The most prominently used approach for the categorization of WSN attacks in the literature distinguishes between attacks based on the layer of the communication architecture which the attack targets. We adopt this approach in this work since it gives a fine-grained view of the aspects of the different algorithms and (or) protocols that each attack seeks to leverage. For each layer of the network model, we first discuss the potential attacks before delving into the different defenses that could be used to thwart the attacks (51).

III.3.1.Physical Layer Attacks

We discuss here several physical layer attacks, including tempering, jamming, and eavesdropping and traffic analysis(51).

III.3.1.1.Tempering

Given physical access to a WSN node, the attacker could temper with the node in several ways, which include reprogramming the node with the aid of easily accessible tools, compromising data stored on the nodes, and the complete physical destruction of a node(51).

III.3.1.2.Jamming

In this type of attack, the adversary sends out signals (e.g., using a specialized waveform generator that interferes with the radio frequencies being used by the WSN. Depending on the specific mechanism used to launch the jamming attack, a sizeable portion of the network could be disrupted, rendering nodes unable to send or receive data along the channel. Figure 9 illustrates a typical jamming scenario where a jammer interferes with communications associated with all nodes within a certain radius, r, of the jammer(51).

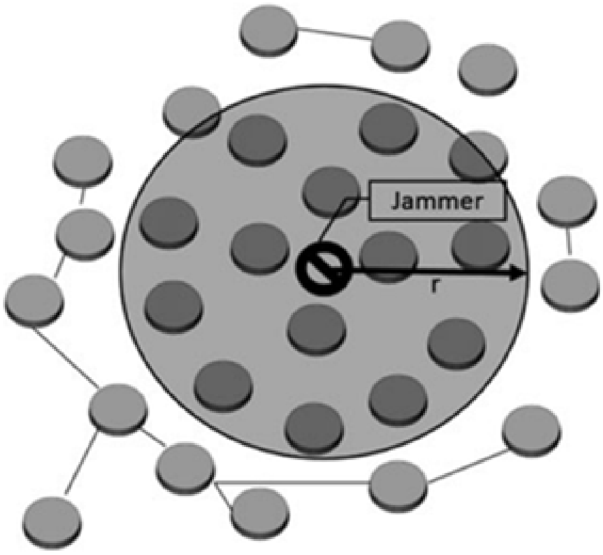


Figure 9. A jamming attack disrupting all communications between nodes within a radius *r* of the jamming node(51).

III.3.1.3.Eavesdropping and Traffic Analysis

Since WSN signals are broadcast in the air, an adversary within the range of the signals could listen to the transactions going on in the wireless channel with the aid of antennas that cost as little as $20. From data captured during eavesdropping, the adversary could directly extract the message content, or carry out traffic analysis to make inferences such as the location of the base station, which could in turn inform more targeted attacks(52).

III.3.2.Physical Layer Defenses

One potential defense against tempering is the design of tamper-proof WSN nodes (e.g., by encasing them in a physically sturdy package. The challenge with this option, however, is that the cost of each mote could tremendously increase, making the deployment of large WSNs very expensive. Under the assumption that an adversary accessing a WSN node will have to move it (e.g., from one place to another), building location awareness into the applications running on the motes could be used to defend against some physical attacks. If a node is determined to have been moved (e.g., based on accelerometer or GPS sensor readings), its data could, for instance, be flagged by other nodes as being potentially compromised, the node could be configured to delete sensitive information from memory if the movement is detected(52).

Jamming and eavesdropping attacks could be mitigated through the use of spread spectrum communication. The philosophy behind spread spectrum communication is to distribute the communication channel over a large number offrequencies, making it very expensive for the adversary to jam or eavesdrop on all the frequencies. In one spread spectrum technique, senders rapidly switch between frequencies using a pattern known to the receivers. This way, an adversary would not know which frequency to jam or eavesdrop while the receiver would be aware of the variations in transmission frequencies. The challenge with spread spectrum communication in broadcast systems such as WSNs is that the adversary only needs to compromise one node to determine the range of frequencies used by the WSN. Also, spread spectrum functionality increases the power requirements of the nodes, and for a large WSN, can dramatically increase the cost of deploying the network.

The ultimate solution to eavesdropping/traffic analysis on the communication channel and tempering attacks aimed to infer information stored on a captured node is encryption/cryptography. Encryption may be applied to the data stored on the motes, or to the packets being exchanged between nodes over the channel. Because cryptography is a solution to a wide range of other attacks(52).

III.3.3.Link Layer Attacks

III.3.3.1.MAC Protocol Violations

MAC protocols generally help ensure that the sensors in the network efficiently use the shared communication channel. When a given node violates the MAC protocol mechanisms (e.g., by sending data during a time slot when another node is supposed to be sending), packet collisions occur. Depending on the extent of the violation, the collisions could result into a wide range of issues, including corruption of the data in the packets, unfair bandwidth usage, and in the worst case, total denial of service if the malicious sender continuously occupies the channel and (or) the attacked nodes continually attempt to retransmit corrupted packets(53).

III.3.3.2.MAC Identity Spoofing

When a node broadcasts data on a WSN, its MAC identity can be accessed by all nodes sharing the communication channel. Given MAC addresses of nodes on a WSN, a malicious node (that may be within the range of the network without necessarily being part of the WSN) could use these identities to masquerade as any of these nodes. If, for example, the malicious node masquerades as an aggregation point, it could leverage this role to access privileged resources of the WSN. The Sybil attack an attack in which a malicious node on the WSN presents different identities to the network at different points in time or cycles through multiple identities to create the impression that they are all simultaneously present on the network is realized in WSNs through MAC identity spoofing(53).

III.3.4.Link Layer Defenses

Error-correcting codes can be employed to address data corruption issues caused by packet collisions if the extent of collision is low to moderate if collisions are comparable to those seen due to probabilistic errors. However, this approach would not handle a heavy volume of collisions as it would require a considerable amount of resources. Incorporating rate-limiting in the MAC protocol to ignore requests beyond a certain threshold of requests and the use of time-division multiplexing to limit the amount of time that nodes can use the channel is the other potential defenses against attacks which overload the channel through violations of the MAC protocol.

MAC identity spoofing can be tackled by associating each identity with a secret key, making it impossible for the adversary to use a given address without having its associated key. Under the assumption of an immobile WSN, position information (registered when the network is set up) can be used in conjunction with MAC identities to confirm that a node is not using a spoofed address. Other techniques, such as node registration (at a central server which can be polled to verify the identities of nodes in the network), and code attestation (e.g., to remotely determine the legitimacy of the node by comparing its memory contents to those of known legitimate nodes) have also been proposed to address these kinds of attacks(54).

III.3.5.Network Layer Attacks

Routing Manipulation Attacks The routing tables used by nodes to forward packets in a WSN can be poisoned in several ways. For example, a malicious node may modify or spoof route update packets before forwarding them to the other nodes in the network. As a result, the nodes receiving these updates may direct traffic along routes determined by the attacker, which could in turn result in congestion and collapse of the network. Some specific types of route-poisoning attacks include:

III.3.5.1.Sinkhole attack

In this attack, the adversary manipulates routing information to lure a large number of nodes into routing their traffic via a node controlled by the adversary (we also alternately refer to this node as the malicious or attacking node)(54).

III.3.5.2.Wormhole attack

This attack is centered on route manipulations designed to make two distantmalicious nodes appear to the other nodes to be much closer to each other than is actually the case. Figure10 illustrates the concept of a wormhole attack. The compromised nodes 11 and 2 have a direct (wormhole) path which is much shorter (in terms of a number of hops) than other potential paths between the two parts of the network. Nodes 12 and 1 are very likely to communicate via the wormhole path since this path creates the illusion that these notes are very close to each other (in fact, much closer to each other than they actually are)(54).

Without the wormhole attack, these two nodes would ordinarily have communicated via the much longer route via nodes 11, 8, 6, 7, 5, 4, and 2(54).

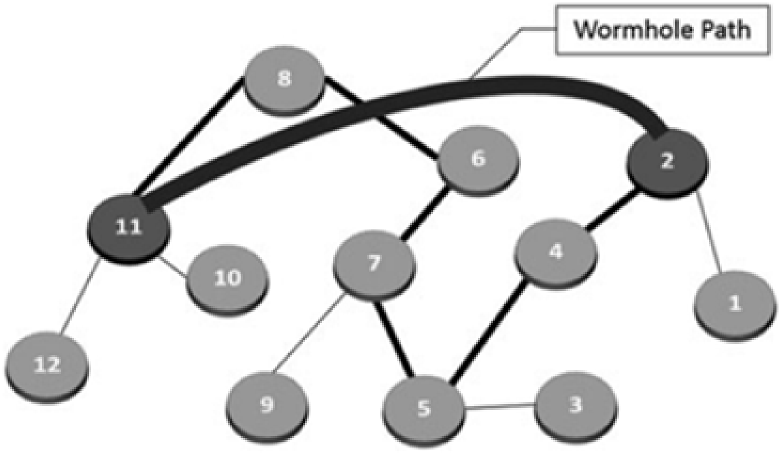


Figure 10. A wormhole attack in WSN(54**).**

If the adversary strategically locates the wormhole in such a way to fool the other nodes that they are just a few hops away from the base station via the wormhole, another malicious node is placed between the wormhole and the base station can then manifest as a sinkhole. Besides the sheer destabilization of the routing process, attacks such as the sinkhole and wormhole attacks can also be used as a vehicle for eavesdropping given the high amount of traffic traversing the nodes being controlled by the adversary(55).

III.3.5.3.HELLO Flood attack

This attack basically involves the attacker sending HELLO packets to various nodes in the network to dupe them into classifying the attacking node as their neighbor. Because some of the nodes may, in fact, be very far away from the malicious node, the adversary may have to use a high-powered transmitter (e.g., laptop class device) to achieve the range of transmission required to deliver the HELLO packets. If this malicious node (now perceived to be a neighboring node) broadcasts a low-cost route to the base station, WSN nodes may attempt to send data via this route, potentially resulting into failed data transfers, retransmissions and channel congestion (since the offending node is actually not in radio range with many of the nodes attempting to send data via it.). Attacks such as the sinkhole and wormhole attacks could be realized with the aid of a HELLO Flood attack(55).

III.3.5.4.Acknowledgment Spoofing

When a node A sends out packets to node B, routing algorithms used by WSNs require that B (explicitly or implicitly) sends some form of acknowledgment to A if data from A is indeed received by B. A malicious node C that is aware of data being sent from A towards B (awareness of the transaction arises out of the broadcast nature of the channel) can spoof the identity of node B and send acknowledgment information towards A, which would then believe that the acknowledgment indeed originated from B. This kind of attack could, for instance, fool node A to believe that node B is alive yet it is in an actual sense dead. With node A having a wrong view of the network topology, an unstable routing process could result(55).

III.3.5.5.Selective Forwarding

Rather than forwarding all received messages as is supposed to be the case in multi-hop networks, a malicious node launching this attack may only forward a subset of the messages. This kind of attack could, for instance, be launched by an adversary who is interested in suppressing the propagation of traffic originating from a certain node or subset of nodes. In the extreme case, a malicious node launching this attack could drop all packets received by it, creating what is termed as a black hole.

Figure 11 illustrates a black hole in the middle of a WSN. The dark-colored nodes route their traffic via the black hole, which drops all packets and denies them access to the base station(56).

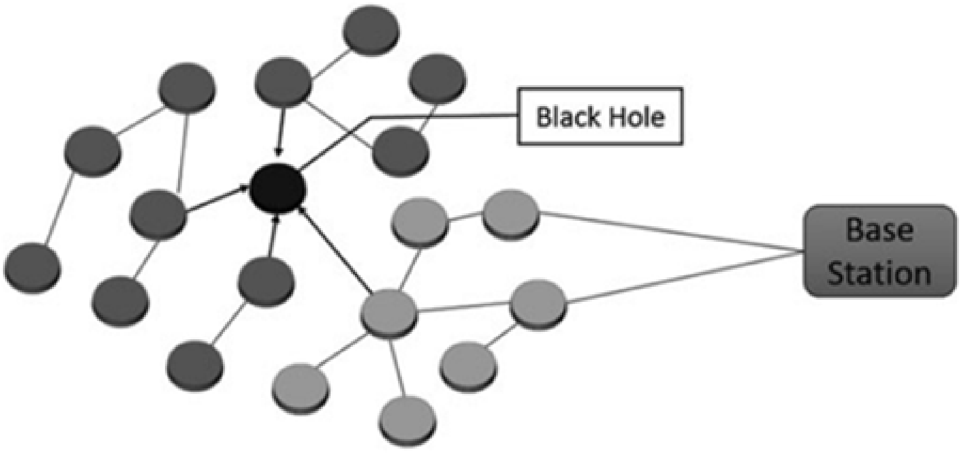


Figure 11. Blackhole illustration: all messages received at the black hole are not forwarded to their destinations(56).

III.3.5.6.Routing Table Overflows

Route-poisoning can also be induced by overflowing the routing tables in the victim nodes. In particular, by continually sending void routing information to the WSN, an adversary could ensure that nodes in the network have bogus routing information in their routing tables, with little or no room available in the node buffers for correct routing information(56).

III.3.6.Network Layer Defenses

The use of multiple paths for data transfer can limit the effects of some route manipulation attacks (e.g., selective forwarding, and the sinkhole and black hole attacks) since the damage to the system would be restricted to only the traffic traversing a subset of the paths. The challenge with this approach, however, is that the maintenance of potentially redundant routes may have its toll on network resources,let alone the scheme not being feasible in sparse networks that have very few possible routes. Authentication schemes that only update routing table entries after the node originating update is verified are another defense against these kinds of attacks. At the message level, a message authentication code (MAC) attached to each message (including routing-specific messages) can be used to determine if routing information could have been altered(57).

SEAD and Ariadne are two secure routing protocols based on endto end authentication. When a node receives a routing update, it will always verify the sender of the update before accepting the update.

Against wormhole attacks, precise clock synchronization between communicating nodes could be leveraged to estimate the distance traveled by each packet and rule out the possibility that a given malicious node is much further than it claims to be. Geographic routing protocols that use knowledge of each node's location information as additional information to determine the source of a given packet can help overcome attacks such as the HELLO floods or acknowledgment spoofing(57).

III.3.7.Transport Layer Attacks

III.3.7.1.Denial of Service

As part of the mechanisms to ensure end-to-end reliability, transport layer protocols in WSNs maintain state information (e.g., information on the status and identity of each active connection). When an adversary opens up a large number of connections in a short time, the amount of information stored about the connections at the concerned node(s) increases tremendously, and in the worst case can deplete all memory at the nodes(57).

III.3.7.2.Connection Desynchronization

In this attack, an adversary sends spoofed messages to one or both nodes involved in a connection, fooling them into requesting for retransmission of missed frames. The attacker manages to force the nodes into this synchronization recovery phase by carefully selecting the sequence numbers and control flags of the spoofed packets. The main effect of this attack is the wastage of energy and memory of the nodes during the continued retransmission requests(57).

III.3.8.Transport Layer Defenses

A possible defense against connection desynchronization attacks is the authentication of all packets being exchanged during a connection. This helps ensure that packets from a malicious sender cannot be confused with those being sent between the genuine nodes.

For the DoS attacks against the transport layer, One proposed solution to this problem is to require that each connecting client demonstrate its commitment to the connection by solving a puzzle. The idea is that a connecting client will not needlessly waste its resources creating unnecessary connections. Given that an attacker does not likely have infinite resources, it will be impossible for him/her to create new connections fast enough to cause resource starvation on the serving node. While these puzzles do include processing overhead, this technique is more desirable than excessive communication(58).

III.3.9.Application Layer Attacks

III.3.9.1.Data Aggregation Distortion

Once data are collected, sensors usually send them back to base stations for processing. Attackers may maliciously modify the data to be aggregated and make the final aggregation results computed by the base stations distorted. Consequently, the base stations will have an incorrect view of the environment monitored by the sensors and may take inappropriate actions(58).

Data aggregation can be totally disrupted if a black hole or sinkhole attacks are launched. In this scenario, no data can reach the base stations(58).

III.3.9.2.Clock Desynchronization

The targets of this attack are those sensors in need of synchronized operations. By disseminating false timing information, the attacks aim to desynchronize the sensors (i.e., skew their clocks).

For example, in IEEE 802.11 (which can be applied to WSNs), nodes are required to be synchronized with the access point. Beacon packets are broadcasted by the access point periodically. The packets contain timing information to be used by nodes for clock adjustment. Attackers can send false beacon packets with wrong timing information. Once nodes adjust their clocks based on the wrong information, they will be out of synchronization with the access point. Although true beacon packets later can bring them back to synchronization, the nodes will oscillate between the two states and be unstable(58).

III.3.9.3.Selective Message Forwarding

For this attack, the adversary has to be on the path between the source and the destination and is thus responsible for forwarding packet for the source. The attack can be launched by forwarding some or partial messages selectively but not others. Note that the attack is different from the other selective forwarding attack in the network layer. To launch the selective forwarding attack in the application layer, attackers need to understand the semantics of the payload of the application layer packets (i.e., treat each packet as a meaningful message instead of a monolithic unit), and select the packets to be forwarded based on the semantics. In comparison, the selective forwarding attack in the network layer only requires attackers to know the network layer information, such as the source and destination addresses. Attackers decide whether to forward packets according to those kinds of information only and therefore operate at coarse granularity (Fig. 12)(58).

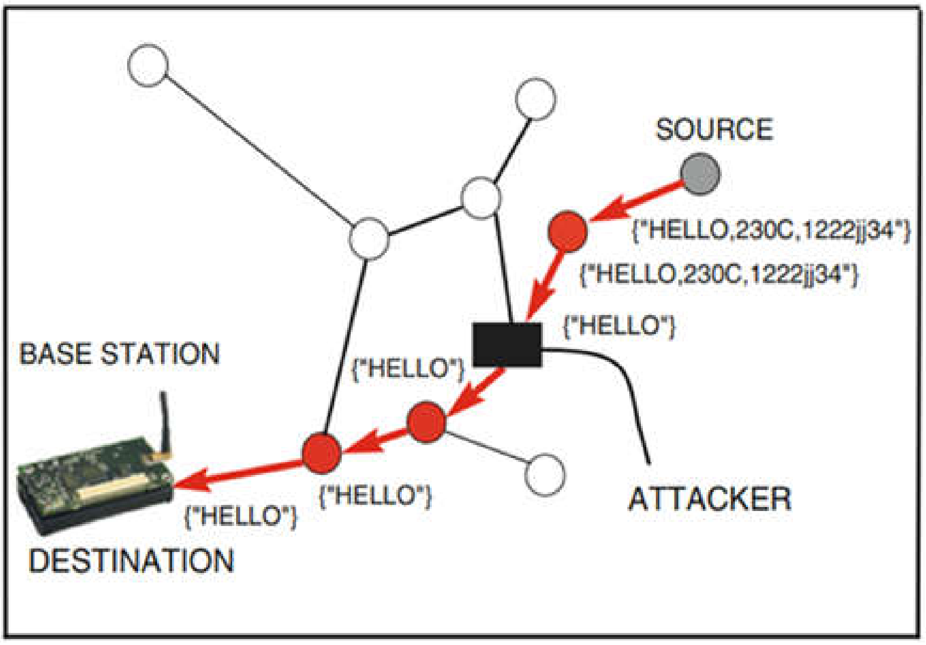


Figure 12. A selective message forwarding example(58)**.**

III.3.10.Application Layer Defenses

For Selective Message Forwarding, we can use Multipath routing. Even if the attacker on one of the paths breaks down the path, the routing is not necessarily broken as other paths still exist(58).
Attacks on data aggregation and clock synchronization are essentially attacks on the integrity of data being exchanged between nodes, meaning they have to be addressed through authentication as data finds its way through the network(58).

An outlier detection algorithm can locate such sensors by comparing their readings with those of their neighbors. In the online deviation detection scheme, an estimation of the data distribution is computed through the input data stream of the WSN. If the current reading of a sensor remarkably deviates from the data distribution (namely the normal readings in the WSN), this sensor will be detected as an outlier. There is also a centralized approach. Base stations launch marked packets to probe certain sensors and try to route packets through them. If a sensor fails to respond, the base stations may conclude that this node is dead(58).

Use authentication methods. With authentication, it can be easily determined whether a sensor can participate in routing or not. Authentication can be either end-to-end or hop-to-hop. In end-to-end authentication, the source and destination share some secret and can thus verify each other.

Hop-to-hop authentication can be combined with multipath routing and result in multipath authentication(58).

III.4.Cryptography in WSN

WSN is set up in a dangerous environment with a broadcasting nature which makes them very vulnerable. WSN security requirement is similar to computer networks.there are many security solutions such as routing security, secure localization, key management, and cryptography.however, due to the limitation of the WSN, they are unable to deal with traditional cryptographic algorithms. The problem involves that the security algorithms presented in a message must be reduced, the sensor consumes energy for every bit it sends which leads to battery life issues. Another problem is the size of the encrypted message, for such a tiny device like the sensor, memory capacity is so limited.

Some cryptographic algorithms were suggested due to its security capabilities that deliver security requirements such as Authentication, Confidentiality, and Integrity.

III.4.1.Types of Cryptographic Techniques

There are a lot of cryptographic methods out there, the security requirement can’t be achieved with the cryptography. In the case of wireless sensor networks, cryptographic methods must be appropriate for the sensor node's constraints such as processing time, data size, and power consumption. So for this to happen, either we adapt what we have or a new technique must be developed. Based on what we have. We can classify the existing techniques into 3:

1. Symmetric Cryptography
2. Asymmetric Cryptography
3. Hybrid Cryptography(67)

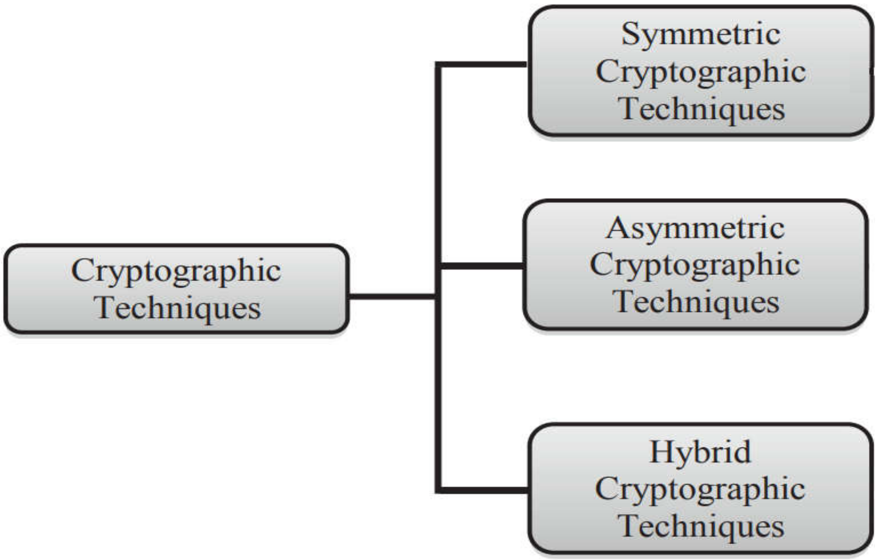

Cryptographic Techniques
Symmetric Cryptographic Techniques
Asymmetric Cryptographic Techniques
Hybrid Cryptographic Techniques

Figure 13. List of Cryptographic techniques**.**

III.4.1.1.Symmetric Cryptography

Symmetric encryption or private key cryptography encrypt and decrypt data using only one key. Let say that A wants to send an encrypted message to B, A will encrypt the message using private key K and B will decrypt the received message with the same key K. The key must remain secret therefore it is transported within a secure channel. Common symmetric encryption algorithms: Data Encryption Standard (DES) and Advanced Encryption Standard (AES).(67)

In this type of cryptography, we are going to explain SPINS, LEAP, and TINYSEC

III.4.1.2.Asymmetric Cryptography

Asymmetric encryption or public-key cryptography encrypt and decrypt data using special keys. For example, the sender uses the public key that belongs to the receiver to encrypt the message, while the receiver uses the private key to decrypt the message. the private key is only known to the receiver while the public key is known to everyone who wants to send a message to the receiver. Common asymmetric encryption algorithms RSA and Elliptic Curve Cryptography (ECC). ECDSA – Elliptic Curve Digital Signature Algorithm and ECDH – Elliptic Curve Diffie Hellman are based on ECC.(67)

Both Symmetric and Asymmetric encryption have their advantages and disadvantages. Symmetric encryption efficiently secures data with low cost, but transporting the private key is dangerous which makes it very compromised. Asymmetric encryption solves that problem, however, comparing to Symmetric it’s slower and uses many more resources.(67)

III.4.1.3.Hybrid Cryptography

Hybrid Cryptography is the combination of the advantages of both symmetric and Asymmetric encryption and avoiding their downsides.(68)

III.4.2.Frameworks of cryptography for WSN

Here we talk about cryptographic frameworks, these frameworks are specially designed for security purposes to the wireless sensor networks. We classify them depending on the key nature whether private or public.(67)

III.4.2.1.Symmetric Cryptographic examples

Here we present frameworks that are design or based on the single shared key for both operations(encryption/decryption).

III.4.2.1.1.SPIN

It is a security building block and it's optimized for resource-constrained environments and wireless communication.(69)

SNEP (Secure Network Encryption Protocol):

It provides slow communication overhead because it only adds 8bytes per message. It also provides data confidentiality which is the basic security primitives that exist mostly in every security protocol, two-party data authentication, and data freshness.
SNEP is a security protocol for sensor networks that uses the CBC-DES (Cipher Block Chaining Data Encryption Standard) encryption algorithm

The results of the encryption of the previous block are fed back into the encryptionof the current block.

The first block of the plain text and a random block of the text called the initialization vector

(IV) is used.it is simply used to make each message unique.

Example:

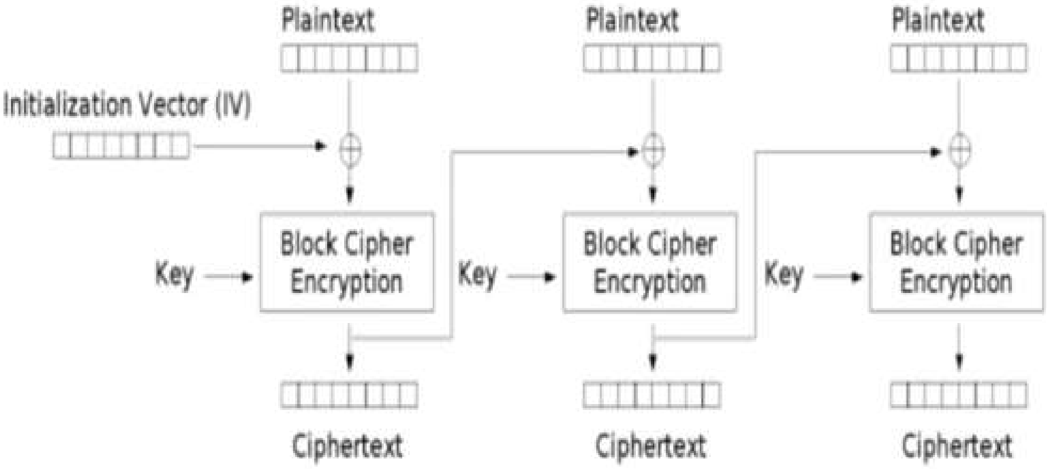


FIG. 14 Cipher Block Chaining (CBC) mode encryption

III.4.2.1.2.LEAP (Localized Encryption and Authentication Protocol)

Is designed for supporting in-networking processing. It establishes 4 types of keys per sensor node :

A. individual key:The individual key is unique for each sensor node to communicate with the sink node. The Individual key is shared between a node and its corresponding base station in order to provide security between them as they communicate(63).

LEAP uses a pre-distribution key to help establish the four types of keys. The individual key is first established using a function of a seed and the ID of the node.

B. pairwise keys: the pairwise shared key is used for secure communications between neighboring nodes. Key shared between a node and its neighboring sensor nodes. cluster keys shared with a set of neighbors(63).

In the pairwise shared key phase, a neighbor discovery process is initiated, and nodes broadcast their IDs. The receiving node uses a function, seeded with an initial key, to calculate the shared key between it and all of its neighbors.

C. group key:The group key is a network-wide key for communication from the sink node to all sensor nodes. A group key, also known as the global key is shared by all the sensor nodes within the network(63).

For distributing the network-wide group key, the sink node broadcasts it in a multihop, cluster-by-cluster manner starting with the closest cluster.

D. The cluster key is used for collaborations within a cluster. The key is shared by a node with multiple of its neighboring sensor nodes. The cluster key is generated by node using a random function and encrypts this key using the pairwise key so that only the authenticated neighbors can decrypt to get access to the cluster key.
The cluster key is distributed by the cluster head using pairwise communication secured with the pairwise shared key(63).

III.4.2.1.3.TinySec

It is a link-layer security architecture, the first fully-implemented protocol for link-layer cryptography in WSN.It provides message authenticity, integrity, and confidentiality. It also provides low bandwidth, latency, and energy cost for sensor network applications.
TinySec implements RC5 encryption algorithms and Skipjack, RC5 is a block designed by Ronald revest in 1994, (RC) stands for revest cipher, revest announced RC2 and EC4 and now there is RC6 witch is AES(59).
Skipjack was initially classified as SECRET so that it could not be examined in the usual manner by the encryption research community(60).
After much debate, the Skipjack algorithm was finally declassified and published by the NSA on 24 June 1998. It used an 80-bit key and a symmetric cipher algorithm, similar to DES(60).
So we are going to explain the steps of AES and DES cryptography:
DES:
1-permutation of the key(to reduce the size).
2-split to left and right.
3-apply the shifting process(normally it’s a left shift).
4-depending on the number of shifts(=s) we end up with 2*s*keys.
5-concatenate the keys (left key number 1 with right key number1 and so on…)we end up with s*keys.
6-permutation 2(another permutation).
7-get the IP of the msg by applying a permutation using the IP matrix than divide it to L0 and R0.
(now we have msg on the form of L0 and R0 and we have s keys)
8-now we have to calculate L1 and R1, L2 and R2, and so on….. to do that we have the rules:
-L(i)=R(i-1)
-R(i)=L(i-1) XOR f(R(i-1),K(i))
-based on that L(1)=R(0) but R(1)=L(0) XOR f(R(0),K(1))
9-we apply E-bit selection(E(R(0))) table on R(0) (permutation).
10- that calculate K(1)+E(R(0)) , K(1) XOR E(R(0)).
11-divide the result to SBOXES.

12-after getting all the SBOXES we then do another permutation and that's the result of the function f.

13-than we XOR the result from f with L(0) to get R(1), and now we have L(1) and R(1).then we continue calculating L(i)and R(i) until i=s (s is the number of subkeys).

14-once we get the L(s) and R(s), we concatenate theme (R(s)+L(s)).

15-than we invert the IP matrix and apply a permutation with it on R(s)+L(s) from the previous step and that's the ciphertext(61).

DES:

1-change the text from block to state(matrix).

The 2-add round key, we XOR the state with the key.

3- subyte , it's a permutation using a given matrix.

4-shift rows, we shift the rows of the matrix starting the shift with 0.1.2.3…… that means we do not shift the first row.

5-mix columns, after shifting we multiplay the result with a matrix (this matrix contain only 01,02,03 these elements) so we have three cases:

*01 the number does not change (0110*01=0110).

*02 here we have two cases

-if the left number is 0 we shift it to the left and that's it (0110*02=1100)

-if the left number is 1we remove that 1 and add a 0 to the right side than XOR with 1B(00011011).
(1100*02=1000 XOR 00011011=00010011)

*03 we multiply it with 02 than 01.

6-after multiplying everything we XOR the results and that's it(62).

III.4.2.2.Asymmetric Cryptographic examples

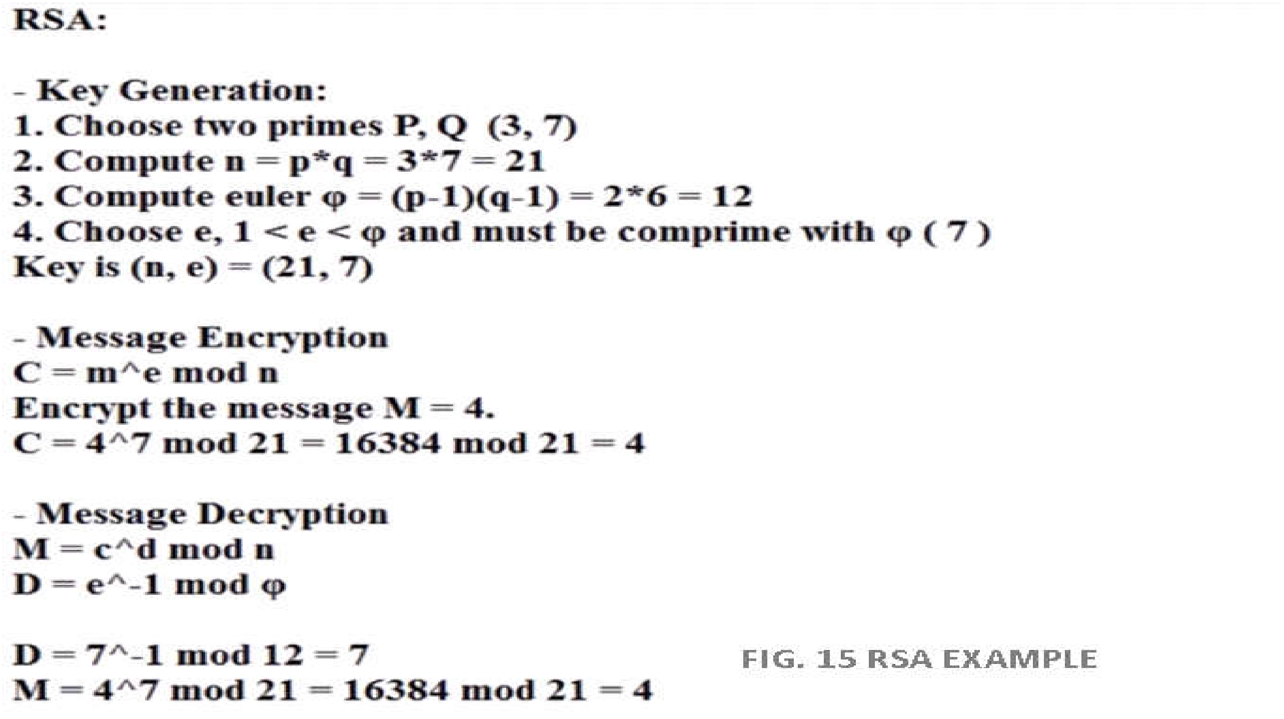
Here we present frameworks that are based on the two shared keys for both private key for encryption and public key for decryption.

We are going to explain RSA and ECC methods.

III.4.2.2.1.RSA

Rivest-Shamir-Adleman or RSA is a Cryptographic algorithm used for security measures to ensure the security for sensitive data. In RSA both keys are used for encryption(private and public) but the opposite key used for the encryption is used for decryption. It provides the confidentiality, integrity, authenticity, and non-repudiation of communications and data storage.(70)

Simple example:



RSA:

- Key Generation:
1. Choose two primes P, Q (3, 7)
2. Compute n = p*q = 3*7 = 21
3. Compute euler φ = (p-1)(q-1) = 2*6 = 12
4. Choose e, 1 < e < φ and must be comprime with φ (7)
Key is (n, e) = (21, 7)

- Message Encryption
C = m^e mod n
Encrypt the message M = 4.
C = 4^7 mod 21 = 16384 mod 21 = 4

- Message Decryption
M = c^d mod n
D = e^-1 mod φ

D = 7^-1 mod 12 = 7
M = 4^7 mod 21 = 16384 mod 21 = 4

FIG. 15 RSA EXAMPLE

III.4.2.2.1.1.TinyPK

Is a mechanism that provides authentication and key exchange between an external party and a sensor network. It’s based on RSA and uses e=3 as a public exponent. TinyPK absorbs the use of TinySec by providing the necessary functionality needed to manually authenticate between a mote and a third-party.(71)

III.4.2.2.2.ECC

Elliptical Curve Cryptography or ECC is a Cryptographic algorithm also used for security measures. An encryption technique that is used for creating faster and smaller, fewer computations when comes to operations and greater efficiency cryptographic keys. This is a great and ideal solution for such tiny devices with energy-constraints like the sensor nodes.(72)

we will explain a little bit how this works:

First, we have the elliptic curve equation that is on the form of **$Y^{2=}X^{3}$+aX+b**and ve have a prime number **P**different a,b,pgive us different elliptic curves. that looks like that:(64)(65)

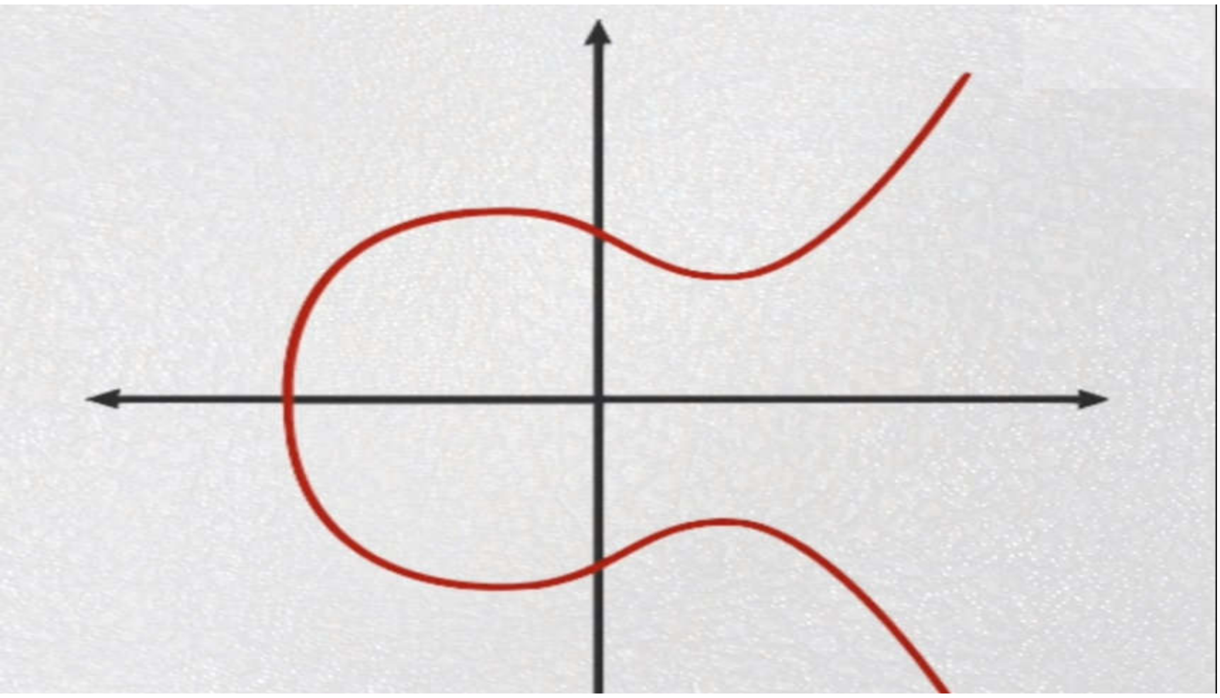


FIG. 16 Elliptical Curve.

To generate a key from this equation we apply what's called the dotting process, where we add two points to get the third point then we repeat the process with the first point and the third point, with repeating this process you will get a cycle of numbers and you will end up coming back to the first point.

So how the dotting process works?

We apply what's called adding the point to itself.

Now we chose any point(G) from the curve and drow a line over it, for example, we use the tangent line.

In the intersection of the tangent with the curve, we mark the point that’s Symmetrical for axisX and that's the result of adding the point to itself(2G)(64)(65).

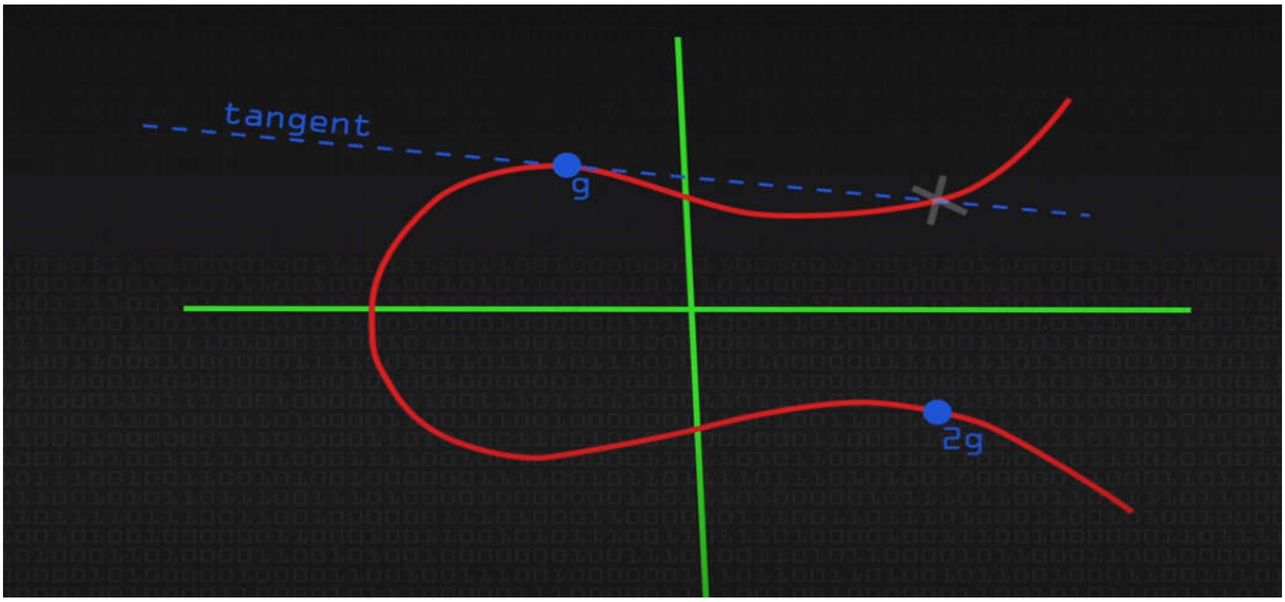


FIG. 17 Dotting in Elliptical Curve(64)(65).

To get (3G) we repeat the process.

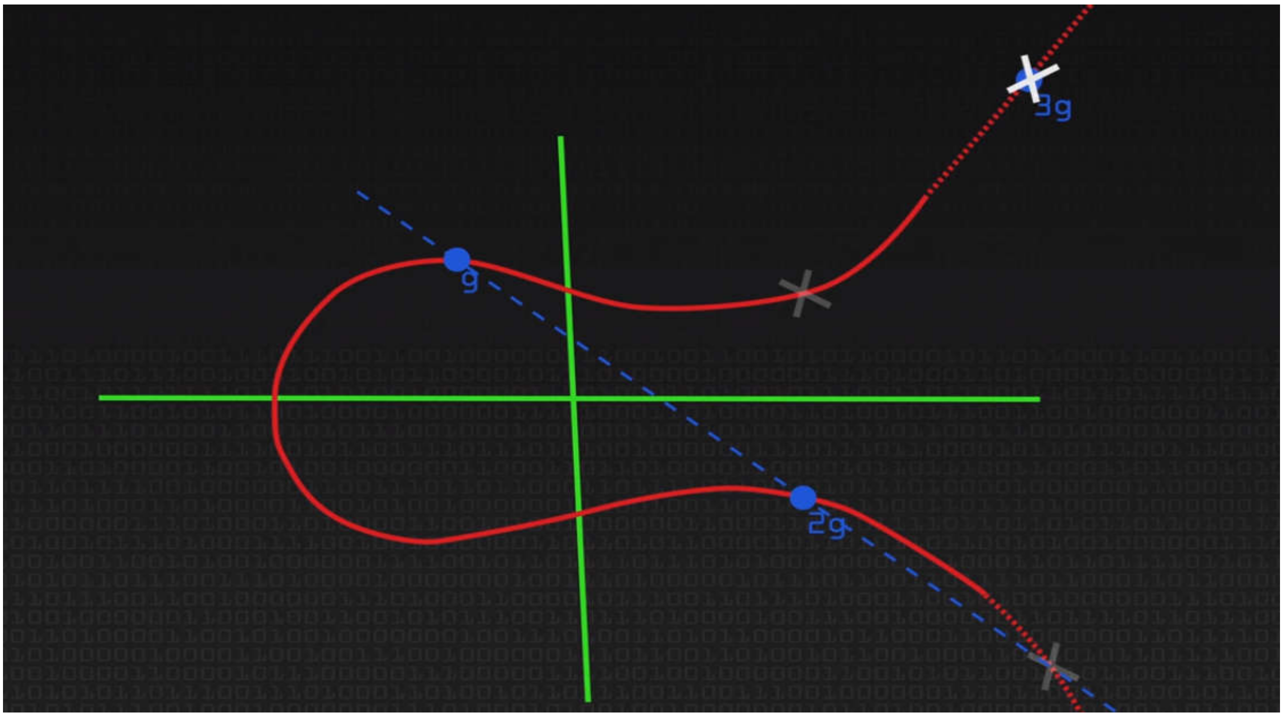


FIG. 18 Dotting in Elliptical Curve(64)(65).

So the idea is if we give you a point in the curve (xG) and we ask you to get x witch is the number of dotting you can not get it because there are millions of choices especially in big elliptic curves and so because of that, that's our private key and that’s the thing that the attacker cant extract back out.
And the key small because we can generate it easier and faster lasts look at a simple example:
We want to get 10G instead of adding G to itself 10 times we simply go like this:
G+G=2G
2G+2G=4G
4G+4G=8G
8G+2G=10G
And just like this, we get 10 G in 4 steps rather than 9steps of adding G to itself(64)(65).

III.4.2.3.Hybrid Cryptography examples:

Here we present hybrid cryptographic frameworks which are the union between symmetric and asymmetric cryptography.
Here we are going to explain MASA algorithms (Mixture of Asymmetric and Symmetric Approaches).-67)

III.4.2.3.1.MASA

The mixture of Asymmetric and Symmetric Approach or MASA is end-to-end data security that reduces resource consumption. It is based on a virtual geographic grid that divides the terrain into regions knows as cells.MASA uses private key sign a hashed event notification to provide security requirements(66).

End-to-End Data Security:

Each node is preloaded with the following parameters

{Km, Kp, Cz, R, B, t}

where Km denotes a master key, Kp denotes the node'sprivate key, Cz denotes the cell dimension, R denotes required confirmation messagesthat should be received from cell members to validate anevent, B denotes the location of the sink, and t denotes thethreshold time that any node waits until it gets a responsefrom the subsequent node along the path to the sink. MASA is divided into the following four phases:(66)

1-Bootstrap Phase: Each sensor computes the center of its cells, l, by using an existing localization scheme. Each sensor then computes the cell key Kc using l and Km,each node broadcasts its existence to its neighbors within the transmission range, each node populates a list of trusted neighbors which consists of node IDs and locations. At the end of this phase, each node deletes its master key to prevent an adversary from deriving the keying material of other nodes(66).

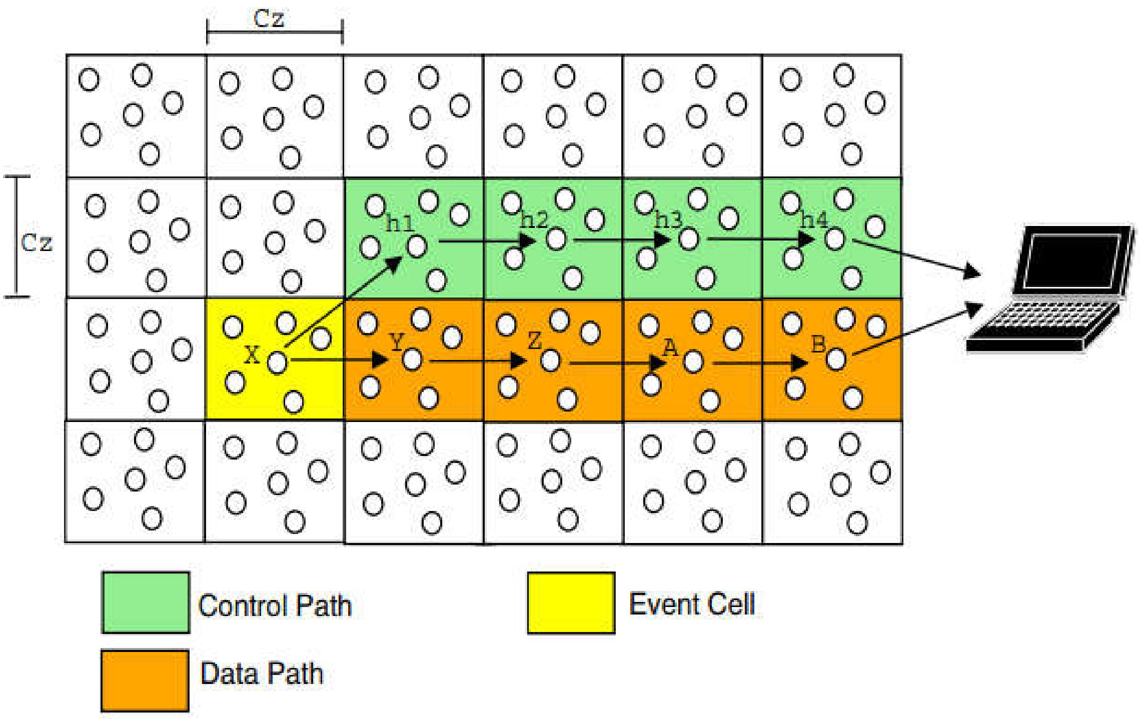


Figure19: First scenario: MASA with no attack

2-Generation Phase: Once a node x (see Figure 2) detects an event, it shares this knowledge with its cell members. A confirmation message is created consisting of:

{S, T ,TimeStamp, EventType}

where S denotes the sender of the event and T denotes the packet's type (includes event, a confirmation message, choosing a helper node, etc). The node encrypts this confirmation message by Kc and broadcasts it. To consider a detected event as a real.

a node should receive R confirmation messages from distinct cell members.

if less than R confirmations are received, then the originator of the original event will be labeled as maliciousand removed from the trusted list.

Once the number of received confirmation messages is ≥ R, then x generates an event message to be sent to the sink as follows:

{S, T, TimeStamp,SI, MD}

where SI denotes Sensitive Information that a node might include in the event message. MD denotes the message digest of the event. The event is digitally signed by applying a one-way hash function to it and then encrypted with theprivate key of x. In addition to this signed part, the source id (S), and event type (T).

x then seeks to establish two paths toward the sink: one for forwarding the event message (a.k.a data path) and the other for monitoring the progress of the event message from one cell to the next (a.k.a control path).

Any existing routing protocols such as AODV could be used for setting up these paths.

The next-hop nodes chosen by x (along the data and control path) must satisfy the following conditions: they must be from its trusted neighbor list, they must be closer to the sink by Cz,and they must be within each others communication range. Figure 2 explains an example where x has chosen y as next hop along the data path and h1 as the corresponding helper node. The function of the helper node is

-oversee the selection of the next hope node chosen by y and ensure that it is one cell closer to the sink.

-the helper node

must overhear the subsequent transmission by the forwarding node along the data path and ensure that the message has not been altered or dropped the helper node must overhear the subsequent transmission by the forwarding node along the data path and ensure that the message has not been altered or dropped.

Once x selects h1, it unicasts a control message asking it to monitor the movement of the event message from y to the next cell as follows:

{S, T, D, TimeStamp}

where D denotes the node that the helper node should monitor which is y. x then starts a timer set for some duration t and unicasts the event message to y. It then switches to promiscuous mode and waits for y to forward the event message to a node in its neighboring cell (node z). If the timer expires and x did not hear the relay transmission from y, y is removed from x's trusted list.

Suppose the location of x is (x1, y1), and the location of z is (x3, y3). Node x ensures that node y forwards the event packet within a certain time t to z such that,

$2Cz \leq Dl \leq 3Cz$

where $Dl = |x3 - x1|2 + |y3 - y1|2$. This ensures that node z is closer than node x to the sink, If z is chosen such that it does not satisfy then a broadcast message is sent by x identifying y as a malicious node(66).

3-Forwarding Phase:

The forwarding phase includes the hop-by-hop forwarding of the event message until it reaches the sink. y repeats what x did in the generation phase except that it does not choose the second helper node h2 which will be chosen by h1, h1 also should be in the radio range of z(66).

4-Verification Phase :

An event is verified at the sink by the signature of the node that created the event. The sink can verify whether the received event is sent by a specific node or not, it calculates the hash code of the event, decrypt the received message digest MD, and then compare the two message digests(66).

III.5.Fault Detection in Wireless Sensor Networks

WSN can break at any time due to its natural composition.breaking points can be manifested in software, hardware, and communication failures, we call this type of vulnerability Faults. The sensor network has 4 levels and failures can appear at each level. Fault awareness is a mechanism that detects, isolates, and mitigates faults at each level. It should be fast and quick to limits the losses but due to the sensor's constraints, it creates a big challenge.(74)

III.5.1. Component level

Revolves the sensor components: sensors, actuators, a power supply, a radio, and a microcontroller. These components are vulnerable to faults because of noise, battery exhaustion, and radio failure. There are some methods to mitigate sensor failures including novel fault-tolerance schemes and detection of faulty sensors using Bayesian algorithms.(74)

III.5.2.Node level

This is where data is processed by converting analog data to digital data. Failure may occur to nodes if they are physically damaged or by any External causes. A fault detection algorithm must be implemented on the sensor, the downside for this method is that it increase the storage but it also saves the communication cost to transmit data.(74)

III.5.3.Network level

« The nodes form a bidirectional communication link with their peers to achieve a network of interacting sensor nodes. The network layer of a WSN is responsible for efficient communication in the transmission medium. Inability to perform efficient routing tasks may constitute to congestion in the network ».(74 page 99)

III.5.4.System level

At this level, data is gathered from all nodes into the base station so it can be transported later on to a local computer or a user with better equipment to analyze the data.(74)

III.Conclusion

Most of these attacks focus on implanting a malicious sensor in the network and convincing other sensors that it belongs to them and that it is one of them.

Therefore, it can be said that focusing on detecting the malicious sensor through strange behavior is a somewhat better solution than focusing on raising the level of encryption, and also developing protocols to detect these malicious nodes maybe even easier.

CHAPTER IV
SIMULATION

In the previous chapter we talked about cryptography used to secure the wireless sensor network, and in this chapter, we are going to implement one of them on a small wireless network.

IV.1.Software

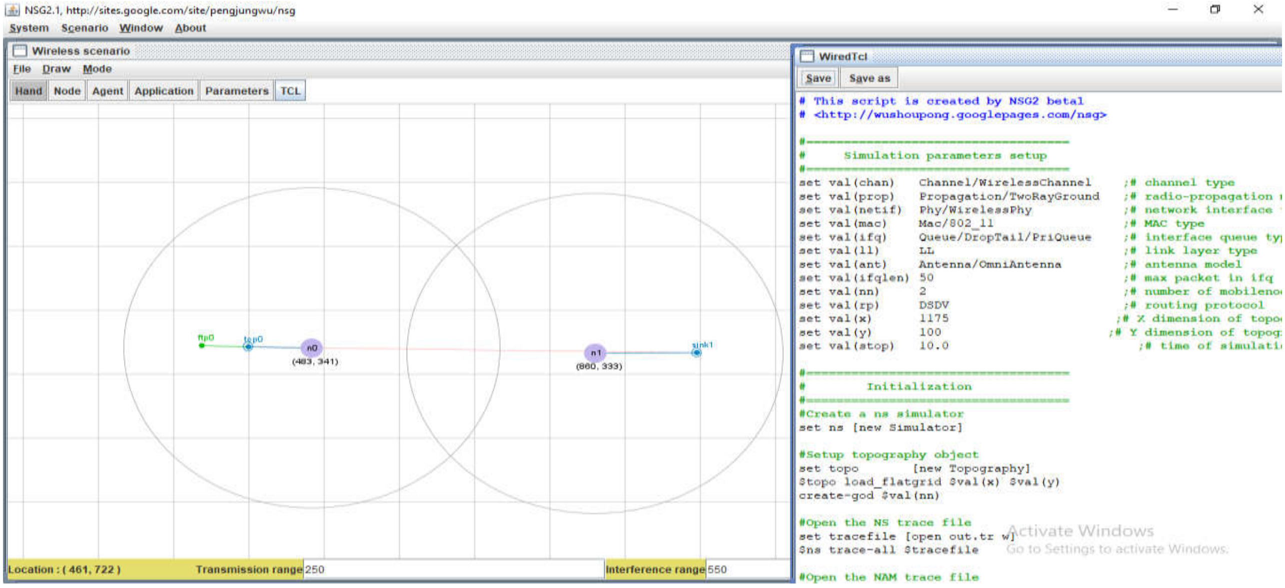
We have simulated the rsa algorithm in a simulation software ns2.Ns2 stands for network simulator version 2 which is an open-source simulator designed for research in computer communication networks. It provides a lot of features such:

1. It is a discrete event simulator for networking research.
2. It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, https and DSR.
3. It simulates wired and wireless network.
4. It is primarily Unix based.
5. Uses TCL as its scripting language.
6. Otcl: Object oriented support.
7. Tclcl: C++ and otcl linkage.(75)

IV.2.Environment

For our simulation environment, we used linux operation system (Linux Mint 19.2 Cinnamon) and we installed ns-allinone-2.35-gcc482 on it.

As we mentioned ns2 uses a special language for simulating wired and wireless networks called TCL.So we need to create a tcl file that contains tcl script to simulate our network. The easiest way to do that is to use a software called NSG2 which is a graphic interface that allows us to create node graphicly and it generates a tcl file for us. Figure 20 shows an example of how to generate a tcl file.

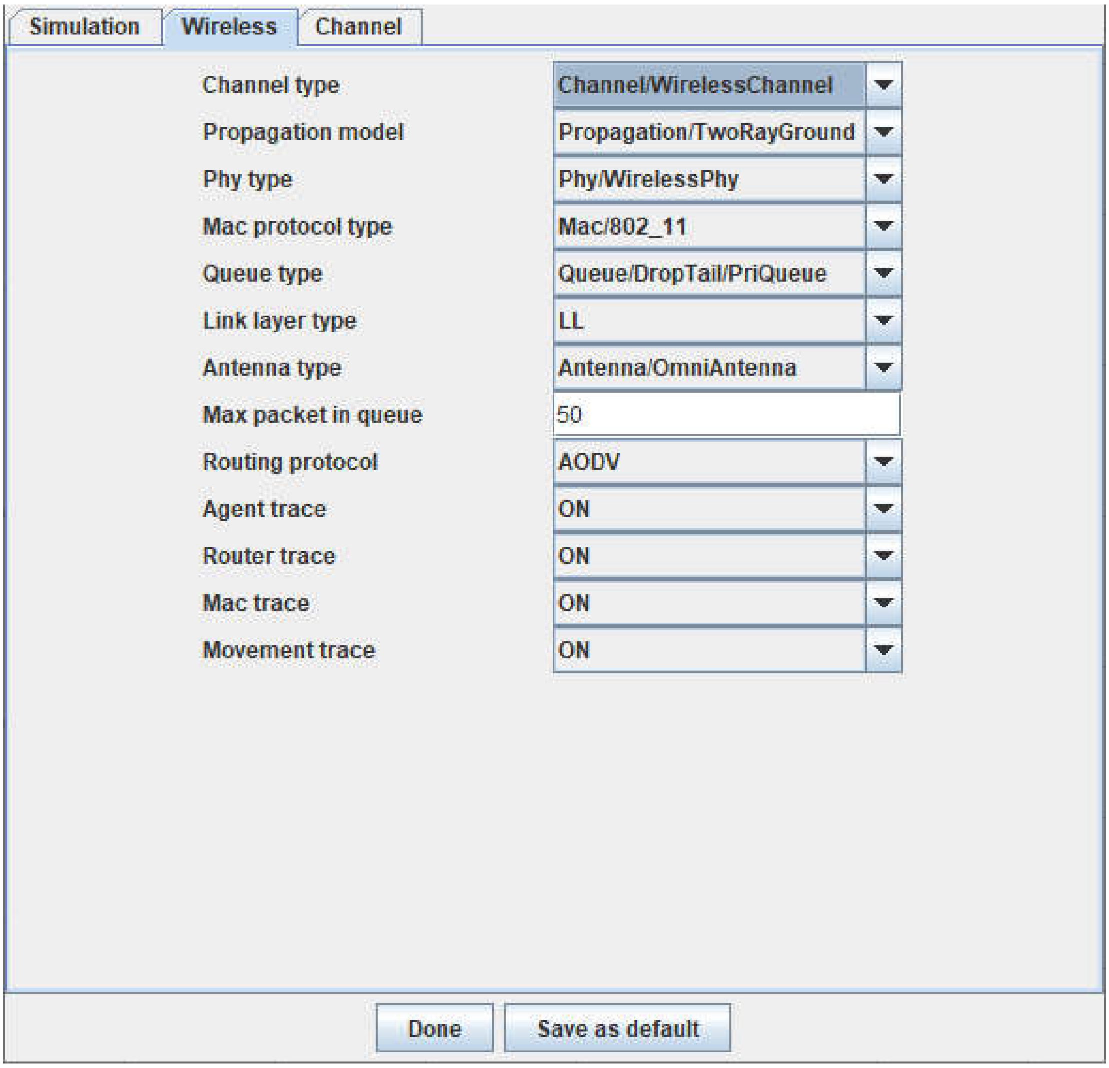


Location : (461, 722)
Transmission range 250
Interference range 550

```
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation 
set val(netif)  Phy/WirelessPhy            ;# network interface 
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue ty
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     2                          ;# number of mobileno
set val(rp)     DSDV                       ;# routing protocol
set val(x)      1175                      ;# X dimension of topo
set val(y)      100                      ;# Y dimension of topog
set val(stop)   10.0                         ;# time of simulati

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
```

FIGURE. 20 Example of creating a wireless network in nsg2.

In our project, we create a simple two nodes wireless network n0 and n1 with n1 been the sink node. Figure 21 shows our network configuration based on nsg2.



Channel type	Channel/WirelessChannel
Propagation model	Propagation/TwoRayGround
Phy type	Phy/WirelessPhy
Mac protocol type	Mac/802_11
Queue type	Queue/DropTail/PriQueue
Link layer type	LL
Antenna type	Antenna/OmniAntenna
Max packet in queue	50
Routing protocol	AODV
Agent trace	ON
Router trace	ON
Mac trace	ON
Movement trace	ON

FIGURE. 21 Example of creating a wireless network in nsg2.

IV.3.Simulation

For us to simulate, in the terminal we use command :ns file.tcl then, nam namfile.nam

Figure 22 shows a screenshot of our network simulation in ns2.

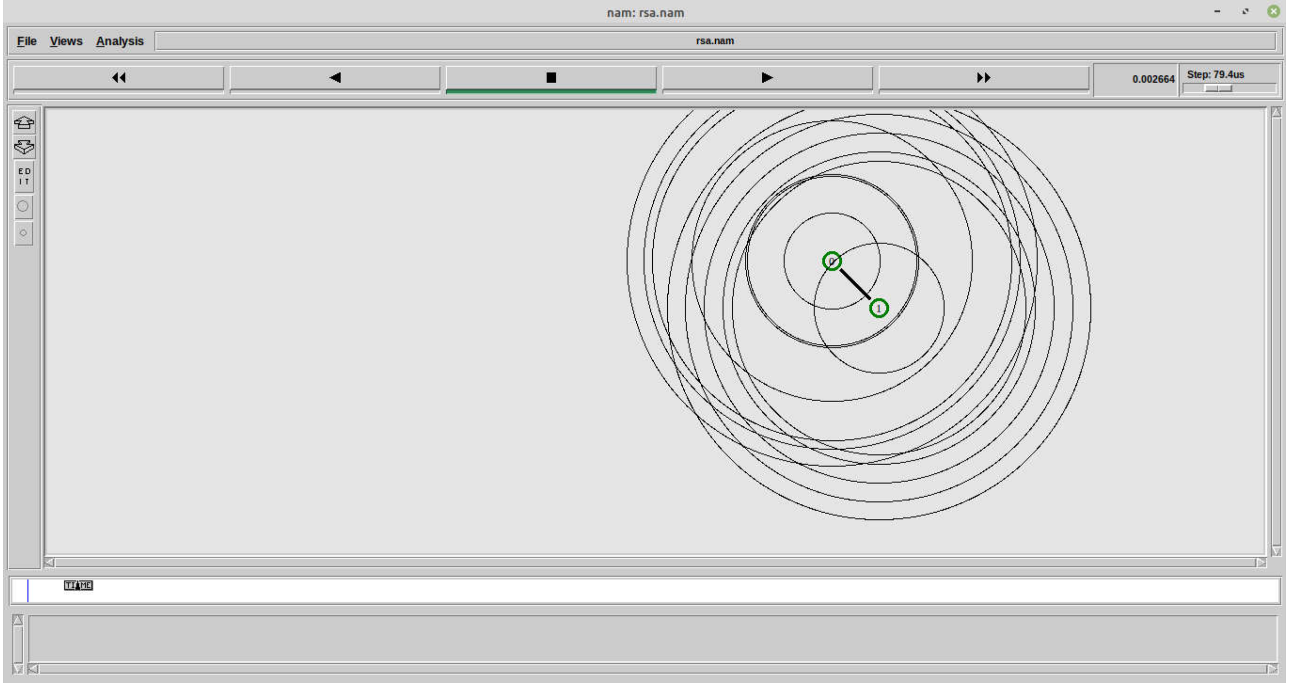


FIGURE. 22 wireless network simulation in ns2.

IV.4.Adding a new protocol

Now all this was done, we need to secure the data transmission in our network. We made a very simple example by sending “hello” message from n0 to n1, encrypted at the source, and decrypted at the destination.Ns2 does not come with a predefined security protocol that we can use so like any other protocol we need to add our security protocol on it. To do so we need to follow these steps:

1. New packet header.
2. C++ code.
3. Tcl code.
4. Some necessary changes(76)(77).

IV.4.1.New packet header

We need to create a new file called security_packet.h:

```
//-------------------------------------------------------------------------------------------------------------
#ifndef ns_security_packet_h
#define ns_security_packet_h

#include "agent.h"
#include "tclcl.h"
#include "packet.h"
#include "address.h"
```

```
#include "ip.h"

struct hdr_security_packet {
    char ret;
    double send_time;
    double rcv_time;    // when security packet arrived
    int seq;            // sequence number
    int data[192];
    unsigned int hashvalue;
    char hedd[192];

    // Header access methods
    static int offset_; // required by PacketHeaderManager
    inline static int& offset() { return offset_; }
    inline static hdr_security_packet* access(const Packet* p) {
        return (hdr_security_packet*) p->access(offset_);
    }
};

class Security_packetAgent : public Agent {
public:
    Security_packetAgent();
    int seq;
    int oneway;         // enable seq number and one-way delay printouts
    virtual int command(int argc, const char*const* argv);
    virtual void recv(Packet*, Handler*);
    void encryption(int asecc[]);
    void decryption(int desc[]);
    unsigned int hashing (char value[], unsigned int len);
};
#endif // ns_security_packet_h
//-------------------------------------------------------------------------------------------------------------
```

IV.4.2.C++ code

Secondly we need to create security_packet.cc

```
//---------------------------------------------------------------------------------------------------------------
#include "security_packet.h"
#include "string.h"

int hdr_security_packet::offset_;

int p=11;



int counter =0;

 int p0,q,n,t,temp[100],j,e,d;

 int decrypttext[100],encrypttext[100],i,len;




void encrypt( int);
void decrypt( int);
 int gcalculated( int a, int b);
 int calculatee();
 int calculated( int);
bool flag;

static class Security_packetHeaderClass : public PacketHeaderClass {
public:
    Security_packetHeaderClass() :
PacketHeaderClass("PacketHeader/Security_packet",sizeof(hdr_security_packet)) {
            bind_offset(&hdr_security_packet::offset_);
    }
} class_security_packethdr;


static class Security_packetClass : public TclClass {
```

```
public:
    Security_packetClass() : TclClass("Agent/Security_packet") {}
    TclObject* create(int, const char*const*) {
          return (new Security_packetAgent());
    }
} class_security_packet;


Security_packetAgent::Security_packetAgent() : Agent(PT_SECURITY_PACKET), seq(0), oneway(0)
{
    bind("packetSize_", &size_);
}


 int calculatee(){
for(i=2;i<t;i++){
   if(gcalculated(i,t)==1)
       return i;
}
return -1;
}

 int gcalculated( int a,  int b)
{
if (b == 0)
 {
  return a;
 }
 else
  return gcalculated(b, (a % b));
}

 int calculated( int x){
    int k = 1;
   while(1){
        k=k+t;
        if(k%x==0)
            return(k/x);
    }

}




//encryption
void encrypt( int e){

     long int pt,ct,key=e,k;
i=0;
len = counter;
while(i!=len){

    pt=decrypttext[i];
    pt = pt -97;
    k=1;
    for(j=0;j<key;j++){
       k = k* pt;
        k = k% n;

    }
    temp[i]=k;
    ct=k+97;
    encrypttext[i]=ct;
    i++;
}
encrypttext[i]=-1;

printf("encrypted message is: ");
printf("\n");
for(i=0;i<len;i++)
    printf("%c",encrypttext[i]);
printf("\n");
}



//decryption
void decrypt( int d){



 int pt,ct,key=d,k;
i=0;

while(encrypttext[i]!=-1){

    ct=temp[i];
    k=1;

    for(j=0;j<key;j++){
       k = k* ct;
        k = k% n;

    }
    pt = k + 97;
    decrypttext[i]=pt;
    i++;
}
printf("decrypted message is");
printf("\n");
decrypttext[i]=-1;
for(i=0;i<len;i++)
    printf("%c",decrypttext[i]);

printf("\n");
```

```
}

int Security_packetAgent::command(int argc, const char*const* argv)
{

if (argc ==3) {

    if (strcmp(argv[1], "send") == 0) {
      // Create a new packet
      Packet* pkt = allocpkt();
      // Access the security packet header for the new packet:
      hdr_security_packet* hdr = hdr_security_packet::access(pkt);
      // Set the 'ret' field to 0, so the receiving node
      // knows that it has to generate an acknowledge packet
      hdr->ret = 0;
      hdr->seq = seq++;
      // Store the current time in the 'send_time' field
      hdr->send_time = Scheduler::instance().clock();
      // copy date to be sent to header

 p0=3;
  q=7;

char hedch[192];
strcpy(hedch, argv[2]);
```

```
    for(i=0;hedch[i]!=NULL;i++){
        decrypttext[i]=hedch[i];
//printf("%c",hedch[i]);
        counter++;
}

   n=p0*q;
   t= (p0-1)*(q-1);

    e=calculatee();




printf("data arg is:");
printf("\n");
int hdrc=0;
while(hdrc<strlen(hedch))
{
printf("%c",hedch[hdrc]);
hdrc++;
}
printf("\n");




char interasc[192];
strcpy(interasc,hedch);
int asc=0;
```

```
hdrc=0;
while(hdrc<strlen(hedch))
{
interasc[hdrc]=hedch[hdrc];
hdrc++;
}



encrypt(e);



asc=0;
for(asc=0;asc<counter;asc++)
{
hdr->data[asc]=encrypttext[asc];
}









      // Send the packet
      send(pkt, 0);
      // return TCL_OK,check if func was called

      return (TCL_OK);
    }
    else if (strcmp(argv[1], "start-WL-brdcast") == 0) {
      Packet* pkt = allocpkt();
      hdr_ip* iph = HDR_IP(pkt);
      hdr_security_packet* ph = hdr_security_packet::access(pkt);
      strcpy(ph->hedd, "test");

      iph->daddr() = IP_BROADCAST;
      iph->dport() = iph->sport();
      ph->ret = 0;
      send(pkt, (Handler*) 0);
      return (TCL_OK);
    }

    else if (strcmp(argv[1], "oneway") == 0) {
      oneway=1;
      return (TCL_OK);
    }
  }


  // call the command() function for the base class
  return (Agent::command(argc, argv));
}



//------------------------------
void Security_packetAgent::recv(Packet* pkt, Handler*)
{
  // Access the IP header for the received packet:
  hdr_ip* hdrip = hdr_ip::access(pkt);

  // Access the security packet header for the received packet:
  hdr_security_packet* hdr = hdr_security_packet::access(pkt);


   if ((u_int32_t)hdrip->daddr() == IP_BROADCAST)
   {
    if (hdr->ret == 0)
    {

      Packet::free(pkt);

      // create reply
      Packet* pktret = allocpkt();

      hdr_security_packet* hdrret = hdr_security_packet::access(pktret);
      hdr_cmn* ch = HDR_CMN(pktret);
      hdr_ip* ipret = hdr_ip::access(pktret);

      hdrret->ret = 1;

      // add brdcast address
      ipret->daddr() = IP_BROADCAST;
      ipret->dport() = ipret->sport();
      send(pktret, 0);
    }
    else
    {
      Packet::free(pkt);
    }
    return;
  }
// end of broadcast mode

  if (hdr->ret == 0)
  {
    // Send an 'echo'. First save the old packet's send_time
    double stime = hdr->send_time;

    char original_data[192];
    int encrypted_data[192];
   // strcpy(encrypted_data,hdr->data);
   // strcpy(original_data,hdr->data);
    int rcv_seq = hdr->seq;

    char out[105];
    unsigned int newhash;
    char authenticate_result[50];
    int deec[192]={0};

    int dcn=0;


dcn=0;
for(dcn=0;dcn<counter;dcn++)
{
encrypted_data[dcn]=deec[dcn];
}


    // Performing decryption

     d=calculated(e);
    decrypt(d);


dcn=0;
for(dcn=0;dcn<counter;dcn++)
{
hdr->data[dcn]=deec[dcn];
}

dcn=0;
for(dcn=0;dcn<counter;dcn++)
{
}
```

```
dcn=0;
for(dcn=0;dcn<counter;dcn++)
{
}





//                    (Scheduler::instance().clock()-hdr->send_time) * 1000);
   Tcl& tcl = Tcl::instance();
   tcl.eval(out);

   // Discard the packet
   Packet::free(pkt);
   // Create a new packet
   Packet* pktret = allocpkt();
   // Access the header for the new packet:
   hdr_security_packet* hdrret = hdr_security_packet::access(pktret);
   // Set the 'ret' field to 1, so the receiver won't send
   // another echo
   hdrret->ret = 1;
   // Set the send_time field to the correct value
   hdrret->send_time = stime;

   hdrret->rcv_time = Scheduler::instance().clock();
   hdrret->seq = rcv_seq;
   strcpy(hdrret->hedd, authenticate_result);//save data to new packet
   // Send the packet back to the originator
   send(pktret, 0);
  }
  else
  {
   char out[105];
    // showing at originator node when packet comes back
//                    (Scheduler::instance().clock()-hdr->send_time) * 1000);
    Tcl& tcl = Tcl::instance();
    tcl.eval(out);
    // Discard the packet
    Packet::free(pkt);
  }
}
//--------------------------------------------------------------------------------------------------------------
```

This file is responsible for the functionalities of the new protocol in our case this is our security protocol. There are two main functions that we take advantage of when they are called when sending and receiving data which are “command” and “recv” respectively.

When command function get called we encrypt data coming from argv[2] which our message “hello” And then we decrypt it once the recv function gets called means that the message reached its destination.

IV.4.3.Tcl code

Now we need to link our protocol to our simulation script :

```
#-----------------------------------------------------------------------------------------------------
#Create two security agents and attach them to the nodes n0 and n2
set p0 [new Agent/Security_packet]
$ns attach-agent $n0 $p0
$p0 set class_ 1



set p3 [new Agent/Security_packet]
$ns attach-agent $n1 $p3
$p3 set class_ 2

#Connect the two agents
$ns connect $p0 $p3
#-----------------------------------------------------------------------------------------------------

#our final tcl file-----------------------------------------------------------------------------------
# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#===================================
#     Simulation parameters setup
#===================================
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     2                          ;# number of mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      731                        ;# X dimension of topography
set val(y)      477                        ;# Y dimension of topography
set val(stop)   10.0                       ;# time of simulation end

#===================================
#        Initialization
#===================================
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open rsa.tr w]
$ns trace-all $tracefile
```

```
#Open the NAM trace file
set namfile [open rsa.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel




$ns use-newtrace



#===================================
#     Mobile node parameter setup
#===================================
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
                -macType       $val(mac) \
                -ifqType       $val(ifq) \
                -ifqLen        $val(ifqlen) \
                -antType       $val(ant) \
                -propType      $val(prop) \
                -phyType       $val(netif) \
                -energyModel "EnergyModel" \
                -initialEnergy 100.0 \
                -txPower 0.9 \
                -rxPower 0.5 \
                -idlePower 0.45 \
                -sleepPower 0.05 \
                -channel       $chan \
                -topoInstance  $topo \
                -agentTrace    ON \
                -routerTrace   ON \
                -macTrace      ON \
                -movementTrace ON
```

```
#===================================
#        Nodes Definition
#===================================
#Create 2 nodes
set n0 [$ns node]
$n0 set X_ 475
$n0 set Y_ 377
$n0 set Z_ 0.0
$ns initial_node_pos $n0 20
set n1 [$ns node]
$n1 set X_ 631
$n1 set Y_ 266
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20



#===================================
#        Agents Definition
#===================================
#Setup a TCP connection
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$ns connect $tcp0 $sink1
$tcp0 set packetSize_ 1500


#===================================
#       Applications Definition
#====================================
#Setup a FTP Application over TCP connection
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ns at 1.0 "$ftp0 start"
$ns at 2.0 "$ftp0 stop"



#====================================
#       Termination
#====================================
#Define a 'finish' procedure
proc finish {} {
   global ns tracefile namfile
   $ns flush-trace
   close $tracefile
   close $namfile
   exec nam rsa.nam &
   exit 0
}



#Create links between the nodes
$ns duplex-link $n0 $n1 10Mb 50ms DropTail

#Set Queue Size of link (n2-n3) to 100
$ns queue-limit $n0 $n1 100

$ns duplex-link-op $n0 $n1 orient right-down
```

```
#Create two security agents and attach them to the nodes n0 and n2
set p0 [new Agent/Security_packet]
$ns attach-agent $n0 $p0
$p0 set class_ 1



set p3 [new Agent/Security_packet]
$ns attach-agent $n1 $p3
$p3 set class_ 2

#Connect the two agents
$ns connect $p0 $p3



#Schedule events
for {set i 1} {$i < 2} {incr i} {
    set result [expr $i /2]
    $ns at $result "$p0 send hello"
}
$ns at 2.0 "finish"



$ns run
#----------------------------------------------------------------------------------------------------
```

IV.4.4.Some necessary changes

We need to make some changes in ns2 so that it can compile our protocol:

IV.4.4.1.Add a new packet type in ~ns-2.35/common/packet.h

```
---------------------------------------------------------------------------------------------------------------
static const packet_t PT_SECURITY_PACKET = 74;                              // Rat mod

      // insert new packet types here
static packet_t      PT_NTYPE = 75; // This MUST be the LAST one

---------------------------------------------------------------------------------------------------------------
```

```
const char* name(packet_t p) const {
        if ( p <= p_info::nPkt_ ) return name_[p];
        return 0;
   }
   static bool data_packet(packet_t type) {
        return ( (type) == PT_TCP || \
              (type) == PT_TELNET || \
              (type) == PT_CBR || \
              (type) == PT_AUDIO || \
              (type) == PT_VIDEO || \
              (type) == PT_ACK || \
              (type) == PT_SCTP || \
                (type) == PT_SECURITY_PACKET || \
              (type) == PT_SCTP_APP1 || \
              (type) == PT_HDLC \
             );
   static packetClass classify(packet_t type) {
        if (type == PT_DSR ||
          type == PT_MESSAGE ||
          type == PT_TORA ||
          type == PT_AODV)
               return ROUTING;
        if (type == PT_TCP ||
          type == PT_TELNET ||
          type == PT_CBR ||
          type == PT_AUDIO ||
          type == PT_VIDEO ||
          type == PT_ACK ||
          type == PT_SECURITY_PACKET ||                       //Rat MoD
          type == PT_SCTP ||
          type == PT_SCTP_APP1 ||
          type == PT_HDLC)
               return DATApkt;
```

```
#define DATA_PACKET(type) ( (type) == PT_TCP || \
                        (type) == PT_TELNET || \
                        (type) == PT_CBR || \
                        (type) == PT_AUDIO || \
                        (type) == PT_VIDEO || \
                        (type) == PT_ACK || \
                        (type) == PT_SCTP || \
                        (type) == PT_SECURITY_PACKET || \
                        (type) == PT_SCTP_APP1 \
                        )
```

IV.4.4.2.Add a new packet type in ~ns-2.35/tcl/lib/ns-default.tcl

```
Agent/Security_packet set packetSize_ 1000;
```

IV.4.4.3.Add a new packet type in ~ns-2.35/tcl/lib/ns-packet.tcl

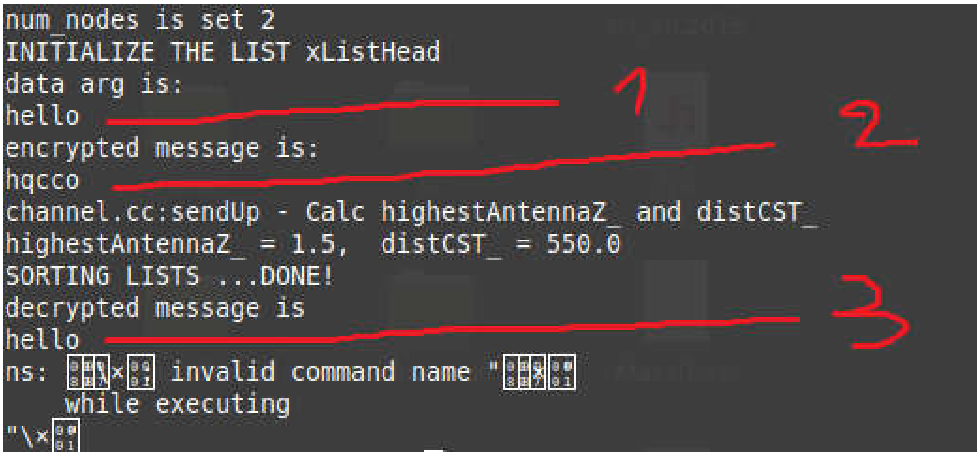
```
# Transport Protocols and related protocols:


   security_packet
.....
```

IV.4.4.4.modify the ~ns-2.35/Makefile.in

```
OBJ_CC = \
.......
   classifier/classifier-port.o src_rtg/classifier-sr.o \
     src_rtg/sragent.o src_rtg/hdr_src.o adc/ump.o \
   qs/qsagent.o qs/hdr_qs.o \
   apps/app.o apps/telnet.o apps/security_packet.o tcp/tcplib-telnet.o \
```

A last we store the two file security_packet.h and security_packet.cc in the app folder. Now we run command ./configure then make, and we good to go.(Note: sometime in case of a mysterious error it’s better to reinstall the ns2 using command: export CC=gcc-4.8 CXX=g++-4.8 && ./install)

Now we can run the command: ns file.tcl. Figure 23 show screenshot for the encryption and the decryption for the message “hello” that get sent from n0 to n1



```
num_nodes is set 2
INITIALIZE THE LIST xListHead
data arg is:
hello
encrypted message is:
hqcco
channel.cc:sendUp - Calc highestAntennaZ_ and distCST_
highestAntennaZ_ = 1.5,  distCST_ = 550.0
SORTING LISTS ...DONE!
decrypted message is
hello
ns: ▯▯▯×▯ invalid command name "▯▯▯▯"
    while executing
"\×▯
```

FIGURE. 23 Encryption and Decryption of a message between two nodes in the wireless network.

GENERAL CONCLUSION

The wireless sensor network is a technology that is constantly achieving great importance in many areas. It consists of sensor nodes that capture data information within its setup environment, process these data, and transmit it via radio signals. For such a tiny device like the sensor the wsn is so limited In terms of memory capacity, battery life, and processing power, that’s why some methods were developed to reduce the resource usages. Also due to its dangerous environment and its broadcasting nature, the wsn is very vulnerable to attacks that could lead to a critical shutdown, and for that reason, Security measures were adapted for protection. We discussed how cryptography can secure data, however, regular cryptography are not quite great for the sensor, so new techniques were developed to be suitable for this type of network.

References

(1)UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE/ALGORITHMES DISTRIBUÉS DE CONSENSUS DE MOYENNE ET LEURS. APPLICATIONS DANS LA DÉTECTION DES TROUS DE COUVERTURE. DANS UN RÉSEAU DE CAPTEURS page 7/Anas HANAF.

(2)Energy-neutral networked wireless sensors page 3/https://www.researchgate.net/publication/260011694_Energy neutral_networked_wireless_sensors.

(3)La Gestion des Données dans les Réseaux de Capteurs sans Fil ,chap1,page 1/Melle HAMOU ALDJA Kahina,Melle HARKATI Samia /Université A/Mira de Béjaïa.

(4)univ Tlemcen/MISE EN OEUVRE DE MECANISMES DE SECURITE BASES. SUR LES IDS POUR LES RESEAUX DE CAPTEURS SANS FIL par Mr SEDJELMACI Sid Ahmed Hichem page 20 .

(5)univ Tlemcen/MISE EN OEUVRE DE MECANISMES DE SECURITE BASES. SUR LES IDS POUR LES RESEAUX DE CAPTEURS SANS FIL par Mr SEDJELMACI Sid Ahmed Hichem page 21.

(6)https://www.youtube.com/watch?v=R462zEQ6RQA

(7)univ Tlemcen/MISE EN OEUVRE DE MECANISMES DE SECURITE BASES. SUR LES IDS POUR LES RESEAUX DE CAPTEURS SANS FIL par Mr SEDJELMACI Sid Ahmed Hichem page 21.

(8)La Gestion des Données dans les Réseaux de Capteurs sans Fil ,chap1,page 3/Melle HAMOU ALDJA Kahina,Melle HARKATI Samia /Université A/Mira de Béjaïa.

(9)La Gestion des Données dans les Réseaux de Capteurs sans Fil ,chap1,page 3/Melle HAMOU ALDJA Kahina,Melle HARKATI Samia /Université A/Mira de Béjaïa.

(10) INTRODUCTION TO WIRELESS SENSOR NETWORKS page 27 /ANNA FORSTER.

(11)Anna Förster - Introduction to Wireless Sensor Networks page 145/146/147.

(12)Anna Förster - Introduction to Wireless Sensor Networks page 145/146/147.

(13)Anna Förster - Introduction to Wireless Sensor Networks page 145/146/147.

(14)Anna Förster - Introduction to Wireless Sensor Networks page 156/157/158/159.

(15) http://tinyos.stanford.edu/tinyos-wiki/index.php/TinyOS_Overview

(16) Anna Förster - Introduction to Wireless Sensor Networks page 16.

(17) Anna Förster - Introduction to Wireless Sensor Networks page 17.

(18)Anna Förster - Introduction to Wireless Sensor Networks page 22,23,24.

(19)La Gestion des Données dans les Réseaux de Capteurs sans Fil /Melle HAMOU ALDJA Kahina,Melle HARKATI Samia /Université A/Mira de Béjaïa page 20,21.

(20)Anna Förster - Introduction to Wireless Sensor Networks page 43,44.

(21)La Gestion des Données dans les Réseaux de Capteurs sans Fil /Melle HAMOU ALDJA Kahina,Melle HARKATI Samia /Université A/Mira de Béjaïa page 23,24.

(22)https://www.actiontec.com/wifihelp/wifi-networking/interference-affecting-wireless-network/

(23)https://itstillworks.com/factors-affecting-wireless-transmission-1029.html By Clare Edwards

(24)https://www.youtube.com/watch?v=J_bf_KE5llQ”

(25)https://www.youtube.com/watch?v=X3ykmJco-kI.6.

(26)https://activereach.net/support/knowledge-base/connectivity-networking/wifi-interference/

(27)https://www.youtube.com/watch?v=zbqrNg4C98U.7-10.

(28) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5982631/ 2018 ”May Jumadi Mabe Parenreng1,2,* and Akio Kitagawa1,* ”

(29)https://prezi.com/2v29aeh49f19/bandwidth-bandwidth-limitations/ by Olusola Odubonojo June 4, 2014

(30)https://www.youtube.com/watch?v=RxaB-KZ_vss

(31)https://www.youtube.com/watch?v=9CqQFhmyYhc.10.

(32) https://www.elprocus.com/architecture-of-wireless-sensor-network-and-applications/

(33) Wireless Sensor Network Security in Military Application using Unmanned Vehicle Arun Madhu1, A. Sreekumar.

(34)https://www.sciencedirect.com/science/article/pii/S1877705812027026. ”Wireless Sensor Network Applications: A Study in Environment Monitoring System Author links open overlay panelMohd FauziOthmanaKhairunnisaShazali”.

(35)https://www.youtube.com/watch?v=kOX9SHiSEl8.12.

(36)https://www.researchgate.net/publication/315794960_Wireless_Sensors_for_Medical_Applications_Current_Status_and_Future_Challenges.12. ”Wireless Sensors for Medical Applications: CurrentStatus and Future ChallengesHadeel Elayan1, Raed M. Shubair 1,2, and Asimina Kiourti 31Electrical and Computer Engineering Department, Khalifa University, UAE2Research Laboratory of Electronics, Massachusetts Institute of Technology, USA3ElectroScience Laboratory, The Ohio State University, Columbus, OH, USA”.

(37) https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.120.46&rep=rep1&type=pdf. ”Wireless sensor networks in precision agriculture Aline Baggio Delft University of Technology – The Netherlands”.

(38)https://www.academia.edu/9144402/Wireless_Sensor_Network_in_Precision_Agriculture_Application. ”Wireless Sensor Network in PrecisionAgriculture Application Mohamed Rawidean Mohd Kassim, Ibrahim Mat, Ahmad Nizar Harun MIMOS, Ministry of Science, Technology and InnovationKuala Lumpur, MALAYSIA”.

(39)https://www.researchgate.net/publication/305657363_Home_Automation_System_using_wireless_Sensor_Network. “HOME AUTOMATION SYSTEM USING WIRELESS SENSOR NETWORKS M.Agalya1, S.Nancy2 and R.Selvarasu3. UG Scholar (1&2), Associate Professor3”.

(40)http://www.mjret.in/V3I1/M11-3-1-1-2016.pdf.4,13. “APPLICATION OF IOT-WSN IN HOME AUTOMATION SYSTEM: A LITERATURE SURVEY Prof S A Jain, Stevan Maineka, Pranali Nimgade Department of Computer Engineering MIT Academy of Engineering, Alandi (D) Pune, India”.

(41)https://www.intechopen.com/books/wireless-sensor-networks-insights-and-innovations/routing-protocols-for-wireless-sensor-networks-wsns-. “Routing Protocols for Wireless Sensor Networks (WSNs) By Noman Shabbir and Syed Rizwan Hassan Submitted: March 13th 2017Reviewed: June 26th 2017Published: October 4th 2017”.

(42)https://link.springer.com/chapter/10.1007/978-3-642-27299-8_11. “Enhanced AODV Routing Protocol for Wireless Sensor Network Based on ZigBee , Dilip Kumar AhirwarPrashant VermaJitendra Daksh”.

(43)https://searchnetworking.techtarget.com/definition/Dynamic-Source-Routing

(44) https://core.ac.uk/download/pdf/158802047.pdf. “Performance Analysis of OLSR Protocol for Wireless Sensor Networks and Comparison Evaluation with AODV Protocol Tao Yang†, Leonard Barolli‡, Makoto Ikeda†, Fatos Xhafa††, Arjan Durresi†‡”.

(45)https://www.researchgate.net/publication/270394473_Optimized_link_state_routing_protocol_OLSRP#pf5. “Optimized link state routing protocol ,Thomas Heide Clausen, Philippe Jacquet”.

(46)https://tools.ietf.org/html/rfc2386#:~:text=QoS%2Dbased%20routing%3A%20A%20routing,for%20a%20duration%20of%20time. “Network Working Group E. Crawley Request for Comments: 2386 Argon Networks Category: Informational R. Nair Arrowpoint B. Rajagopalan NEC USA H. Sandick Bay Networks August 1998”.

(47)Rastko R. Selmic, Vir V. Phoha, Abdul Serwadda (auth.) - Wireless Sensor Networks_ Security, Coverage, and Localization-Springer International Publishing (2016) page 83

(48)Rastko R. Selmic, Vir V. Phoha, Abdul Serwadda (auth.) - Wireless Sensor Networks_ Security, Coverage, and Localization-Springer International Publishing (2016) page 84

(49)Rastko R. Selmic, Vir V. Phoha, Abdul Serwadda (auth.) - Wireless Sensor Networks_ Security, Coverage, and Localization-Springer International Publishing (2016) page 85

(50)Université Hadj Lakhder - Batna Protocole de sécurité Pour les Réseaux de capteurs Sans Fil by Samir ATHMANI 15/07/2010 page 37

(51)Rastko R. Selmic, Vir V. Phoha, Abdul Serwadda (auth.) - Wireless Sensor Networks_ Security, Coverage, and Localization-Springer International Publishing (2016) page 86

(52)Rastko R. Selmic, Vir V. Phoha, Abdul Serwadda (auth.) - Wireless Sensor Networks_ Security, Coverage, and Localization-Springer International Publishing (2016) page 87

(53)Rastko R. Selmic, Vir V. Phoha, Abdul Serwadda (auth.) - Wireless Sensor Networks_ Security, Coverage, and Localization-Springer International Publishing (2016) page 88

(54)Rastko R. Selmic, Vir V. Phoha, Abdul Serwadda (auth.) - Wireless Sensor Networks_ Security, Coverage, and Localization-Springer International Publishing (2016) page 89

(55)Rastko R. Selmic, Vir V. Phoha, Abdul Serwadda (auth.) - Wireless Sensor Networks_ Security, Coverage, and Localization-Springer International Publishing (2016) page 90

(56)Rastko R. Selmic, Vir V. Phoha, Abdul Serwadda (auth.) - Wireless Sensor Networks_ Security, Coverage, and Localization-Springer International Publishing (2016) page 91

(57)Rastko R. Selmic, Vir V. Phoha, Abdul Serwadda (auth.) - Wireless Sensor Networks_ Security, Coverage, and Localization-Springer International Publishing (2016) page 92

(58)Rastko R. Selmic, Vir V. Phoha, Abdul Serwadda (auth.) - Wireless Sensor Networks_ Security, Coverage, and Localization-Springer International Publishing (2016) page 93

(59)https://fr.slideshare.net/serngawy/rc4rc5 20/08/2020

(60)https://www.cryptomuseum.com/crypto/usa/skipjack.htm 20/08/2020

(61)https://www.youtube.com/watch?v=FLszAz7gRqM

(62)https://www.youtube.com/watch?v=GOLN3h-M9GA

(63)Localized Encryption and Authentication Protocol for Secure Key Management in Wireless Sensor Networks

Arundhati Nelli 1, Sushant Mangasuli 2 Manasa N 3

Assistant Professor, Dept. of CSE, Alva's Institute of Engineering and Technology, Mijar, Karnataka, India 1

(64)https://www.youtube.com/watch?v=muIv8I6v1aE

(65)https://www.youtube.com/watch?v=NF1pwjL9-DE

(66)End-to-End Data Security in Sensor Networks Using a Mix of Asymmetric and Symmetric Approaches by Hani Alzaid and Manal Alfaraj

(67)https://www.sciencedirect.com/science/article/pii/S2212017312006640. "Security Frameworks for Wireless Sensor Networks-Review☆ Author links open overlay panelGauravSharmaSumanBalaAnil K.Verma".

(68)https://www.sciencedirect.com/science/article/pii/S2314717215000616. "Two-phase hybrid cryptography algorithm for wireless sensor networks Author links open overlay panelRawyaRizkYasminAlkady".

(69)http://users.ece.cmu.edu/~adrian/projects/mc2001/mc2001.pdf. "SPINS: Security Protocols for Sensor Networks———————————————— Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, J. D. Tygar Department of Electrical Engineering and Computer Sciences University of California, Berkeley".

(70)https://searchsecurity.techtarget.com/definition/RSA." Posted by: Margaret Rouse".

(71)https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.6742&rep=rep1&type=pdf." TinyPK: Securing Sensor Networks with Public Key Technology Ronald Watro, Derrick Kong, Sue-fen Cuti, Charles Gardiner, Charles Lynn1 and Peter Kruus BBN Technologies 10 Moulton St Cambridge MA 02138 617-873-3200".

(72)https://searchsecurity.techtarget.com/definition/elliptical-curve-cryptography." Posted by: Margaret Rouse".

(73)https://link.springer.com/chapter/10.1007/978-3-642-00224-3_19." secFleck: A Public Key Technology Platform for Wireless Sensor Networks Wen HuPeter CorkeWen Chan ShihLeslie Overs".

(74)Wireless Sensor Networks: Security, Coverage, and Localization By Rastko R. Selmic, Vir V. Phoha, Abdul page 98.

(75) https://www.tutorialsweb.com/ns2/NS2-1.htm.

(76) How to Add a New Protocol in NS2 Xu Leiming CSCW Lab. at CS Dept., Tsinghua Univ. mailto:xlming@csnet4.cs.tsinghua.edu.cn June 1, 2001.

(77)https://www.nsnam.com/2015/03/security-protocol-packet-in-ns2.html.