A COMPREHENSIVE ANALYSIS AND HARDWARE IMPLEMENTATION OF COMSEC AND TRANSEC FOR WIRELESS SENSOR NETWORKS

MCS

by

Muhammad Shahzad

A thesis submitted to the faculty of Information Security Department Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment of the requirements for the degree of MS in Electrical Engineering (Telecom)

JUNE 2018

ABSTRACT

A COMPREHENSIVE ANALYSIS AND HARDWARE IMPLEMENTATION OF COMSEC AND TRANSEC FOR WIRELESS SENSOR NETWORKS

By

Muhammad Shahzad

Wireless Sensor Networks (WSNs) require secure communication framework because of their vulnerability against different types of malicious attacks. The main objective of this research work is to implement a secure and reliable communication framework for WSNs. A comprehensive study of existing communication security and transmission security methods for WSNs has been carried out. Encryption algorithm AES-256 has been implemented on microcontroller hardware and modifications have been made to enhance the speed of processing and reduce requirement of computational resources. For transmission security, a mechanism of frequency hopping and its synchronization has been proposed and implemented to achieve hop rate of 100 hops/sec for WSNs. The spectrum of proposed system has also been analyzed using Universal Serial Radio Peripheral 2 devices. Proposed system offers data confidentiality, reliability and improved link quality.

DECLARATION

I hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification either at this institution or elsewhere.

Muhammad Shahzad

DEDICATION

“In the name of Allah, the most Beneficent, the most Merciful"

I dedicate this thesis to my teachers and parents, who helped me each step of the way.

ACKNOWLEDGMENTS

All praises to Allah for the strengths and His blessing in completing this thesis.

I would like to convey my gratitude to my supervisor, Col Adnan Rashdi, PhD, for his supervision and constant support. His invaluable help of constructive comments and suggestions throughout the experimental and thesis works are major contributions for the success of this research. Also, I would thank my committee members; Lt Col Hasnat Khurshid, PhD and Lt Col Usman Mahmood Malik for their support and knowledge regarding this topic.

Table of Contents

LIST OF FIGURES

Introduction

This thesis has three main objectives. First, a comprehensive analysis of communication security (COMSEC) and transmission security (TRANSEC) for Wireless Sensor Networks (WSNs) to find loopholes from where a malicious attack would be launched. Secondly, to modify existing COMSEC to enhance the speed of processing and reduce the amount of resources on microcontroller. Last one is the TRANSEC part which includes the design and implementation of frequency hopping mechanism and its synchronization to achieve hop rate of 100 hops/sec. for WSNs on low cost microcontroller. Section 1.1 describes overview of WSNs. In Section 1.2, motivation of our work is presented. In Section 1.3, Security issues of WSNs are described. In Section 1.4, purpose of security in WSNs is described. In Section 1.5, constraints of WSNs are discussed. In Section 1.6, Advanced Encryption Standard AES-256 is discussed. Section 1.7 describes frequency hopping technique. Problem statement of this thesis is described in Section 1.8. In Section 1.9, objectives of this research are presented. Finally, chapter is concluded by describing the outline of structure.

1.1 Overview

Self-configured and infrastructure less wireless systems to observe natural or physical conditions are known as Wireless Sensor Networks (WSNs) [1]. Examples of these conditions include, sound, temperature, movement weight and vibration. Analog signals are captured by sensor nodes and then converted to digital signals in order to process in microprocessor. Base station behaves as interface between clients and networks. Tolerance and scalability are the most important features of WSNs [2]. Due to these features, they have their applications in various fields, for example remote healthcare monitoring [3], agricultural monitoring system [4], environmental monitoring system [5] and fire monitoring [6]. In general, a WSN is comprised of a large number of sensor nodes. These sensor nodes use radio signals to convey information among themselves. A WSN has a power unit, a transceiver, a microprocessor and sensing unit. The individual nodes in a Wireless Sensor Network (WSN) naturally has limited resources: they have constrained processing speed, less transmission bandwidth and limited storage. After the deployment of sensor nodes, nodes has to self-organize themselves. When setup is completed, sensors start to collect data from environment. Specific tasks are performed as instructed from control unit. Sensors can be operated in continuous, periodic or event oriented. The details of these events is described in detail in later section of this thesis.

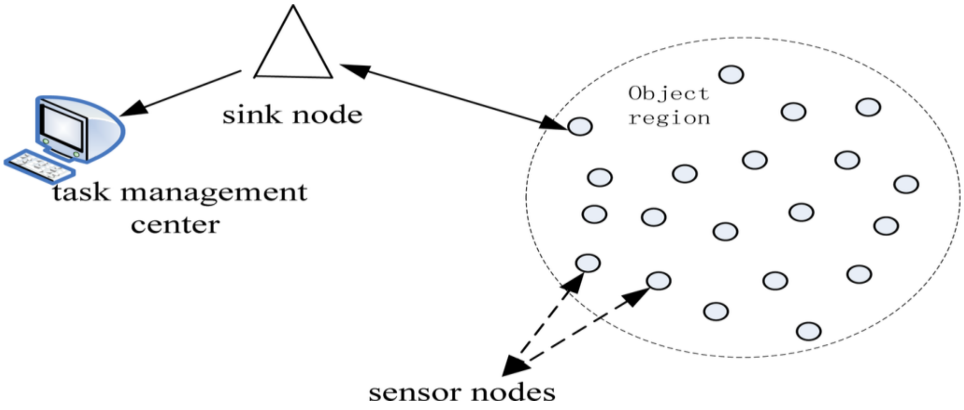


Figure 1.1: General form of Wireless Sensor Network

1.2 Motivation

WSNs have their applications in battlefield monitoring, medical care, surveillance and environmental monitoring. These applications often require that communication must be secure, efficient, fault tolerant and tamper resistant. In WSNs, base station gathers data from nodes. Attackers can observe data and modify, interrupt or fabricate data and can inject wrong information to sinks or base stations. So, a powerful COMSEC with efficient resource utilization is required to overcome these attacks to provide security in WSNs. Along with COMSEC, TRANSEC is also of great importance. For this purpose, frequency hopping is a spread spectrum technique which uses multiple channels to communicate across the network. It mitigates the effect of jamming and interference. Frequency hopping is promising to achieve coexistence with currently deployed wireless systems. It is successful though with limited computational resources of WSNs.

The principle motivation behind executing a cryptographic algorithm in WSNs is to ensure the protection of their data. A Wireless Sensor Network organize works by utilizing small sensors in remote zones. These sensors help in observing physical or natural elements and provide communication among themselves. They gather important required information, process them and send back the required data to the base station. Calculations in view of public key cryptography and private-key cryptography assume an important part in the security of data. The most essential part of information insurance is to analyze different sensor networks applications and apply the most reasonable cryptographic algorithm. The most broadly utilized Private-Key Cryptography calculation is Advanced Encryption Standard (AES) is a block cipher. It is the most generally utilized encryption algorithm. It is quicker and efficient in processing as opposed to Public-Key Algorithm.

1.3 Security Issues in WSNs

For some, wireless sensor networks (WSN) in mechanical, military applications and medical, secure activity is compulsory [7]. The requirements of security in WSNs are somewhat unique in relation to the normal necessities of basic communication systems. In every one of these systems, the security is portrayed by authenticity, data confidentiality, replay prevention and integrity [8]. Information privacy guarantees that the information transmitted by a node don't break to neighboring systems. The standard way to deal with confidentiality is by using data encryption. To restrict insertion of malicious data by an enemy, authentication is used. It causes the beneficiary to affirm that the got information are sent from a trusted source. Secure hash functions are implemented to provide data integrity. It must be ensured by using reply prevention that no one could bypass security schemes. Integrity can be given by actualizing secure hash capacities. The specific properties of Wireless Sensor Networks, i.e., harsh environment, network size, changing topology and dense deployment, make the system security simple to bargain if suitable prevention schemes are not used.

1.4 Purpose of Security in WSNs

WSNs exhibit some characteristics which are unique to them. Main purpose of security services of WSNs is to protect communicated data from malicious attacks. Data confidentiality is basic requirement of security services. It ensures that only recipient should understand the message in a network. Data confidentiality guarantees that readings of sensor node are not accessible to un-authorized neighbors, mechanism of key exchanging must be robust and public information must also be in encrypted form to protect malicious attacks. Data integrity ensures that message is not altered by adversary. Another purpose of security services is that in un-fair conditions i.e. in presence of attack, communication should not break. Different approaches are followed to overcome this problem. Some of them require additional communication links among nodes. Master client model is also used to serve the matter. Data freshness guarantees that old messages are not retransmitted by adversary and information is latest. This attack is more feasible in shared-key cryptography where adversary uses previous key to launch reply attacks. Time counters can be added to check data freshness. Self-healing and self-organizing is also required by WSNs and it poses a lot of challenges in security. Secure localization is an important purpose of security. In many cases, accurate location of sensor node is difficult to find in WSNs. Adversary can cheat the network by providing wrong location information by reply messages and wrong signal strength. Time synchronization is also required by many applications of WSNs. So security mechanism must be time synchronized. Authentication confirms that sensor node is the node which it should be. It must be verified that transmitted data is from authenticated source. Message authentication codes can be used to overcome this issue.

1.5 Wireless Sensor Networks Constraints

WSNs consist of huge number of sensor nodes and these devices are constrained in resources. These sensor nodes have limited capability of processing, limited bandwidth and low capacity of storage. Physical size of sensor nodes, low cost and limited energy are the main causes of these constraints. Conventional security algorithms cannot be directly implemented on WSNs because of these constraints. So to implement security algorithms, awareness of nodes is necessary that how much resources are available. Memory is usually limited in WSN nodes. Typically a node contains SRAM and flash memory. Flash memory contains firmware and code of application. It can be used to store sensor data and results of computation. RAM is used to store data for processing. Usually RAM is limited and much less than ROM but it is faster than ROM. So a trade is there to optimize between RAM and ROM. After loading code of application and program, only limited amount of memory resources to place complicated algorithms. Energy is also a big constraint in WSNs. As sensor nodes are located in remote areas and power supply is difficult to provide. So tiny batteries are used to provide power. Energy consumption of nodes is divided in three parts. First, Sensor transcoder energy, secondly, microcontroller processing energy and at last, transceiver energy. Each transmitted bit in WSNs execute about 1000 instructions and consumes much amount of power. Any calculation of cipher costs a lot of power consumption. So as security level of system is increased, power consumption is also increases. Another serious threat in WSNs is unreliability of communication. Commonly, connectionless protocols based routing and hence unreliable. At highly congested nodes and due to channel errors, packet may drop. Unreliable channel may damage the data packets. Even if channel is good, communication may not be as good because wireless communication has broadcast nature. Latency of communication is another constraint of WSNs. Network congestion, multi hop routing and processing for networking causes latency in transmission of packets. To achieve synchronization in this case is very difficult especially if we are using spread spectrum techniques.

1.6 Advanced Encryption Standard (AES)

The U.S. government selected The Advanced Encryption Standard (AES) as standard to secure data. This algorithm is implemented for both hardware and software platforms in all over the world to provide data security. Due to the loop holes present in DES, the National Institute of Standards and Technology (NIST) started the development of Advanced Encryption Standard in 1997. It was proposed by keeping in mind that it must be easily implementable on hardware and software platforms with low computation complexity and should be able to provide high data security and reliability. Effective use of AES by U.S. government and private institutions, make the AES most powerful symmetric key cryptography algorithm. AES provides better security as compared to 3DES and DES, which makes it perfect for fast computing devices with high throughput i.e. switches and firewalls.

1.7 Frequency-Hopping Spread Spectrum (FHSS)

During radio transmission, repeated switching of frequencies is made to reduce interception and avoid interference, and this technique is called Frequency-hopping spread spectrum (FHSS). It is valuable to obstruct jamming, or eavesdropping of broadcast communications. Along with these, it can limit the impacts of interference. In FHSS, the transmitter jumps between accessible narrowband frequencies inside a pre-determined narrowband channel in a pseudo random manner known to both receiver and transmitter. A short amount of information is transmitted on the ongoing narrowband channel, at that point receiver and transmitter tune to the next frequency known to both of them. In general, the transmitter will jump to another hope many times every second. Chances of some other transmitter being on a similar channel in the meantime are low because no channel is utilized for long. FHSS is frequently utilized as a technique to permit different receivers and transmitters to work in a similar space on a similar channel in the meantime.

1.8 Problem Statement

In un-trusted and hostile environments, like battlefield, traffic can be eavesdropped by an adversary. They can inject wrong messages. Therefore appropriate secure mechanisms needs to be in-corporate in WSNs. Due to the constraints on bandwidth, energy consumption and processing, it is very difficult to provide COMSEC and TRANSEC in WSNs. For example ATmega328 consists of 16 MHz clock with 1 KB of EEPROM and 2 KB of SRAM. It leaves very limited computational resources to provide communication and transmission security in WSNs. We cannot deploy traditional security protocols without modifications due the hardware limitations. For example, the Diffie-Hellman algorithm [9] and RSA [10] which are asymmetric cryptography algorithms, are expensive to use in WSNs due to their computational complexity.

In WSNs, TRANSEC is also of great importance. While working in unfriendly conditions, if the WSNs remain on a single channel, their capacity to resist the jamming and interference will be poor. What's more, the loss of data will be increased if jamming is applied on fixed channel and quality turns out to be low. So the settled channel can't be utilized when data is important and needs are high. A precise work is required to design implement hopping algorithm for WSNs due to the limited hardware resources.

1.9 Objectives

A strongly redundant and reliable system against malicious attacks, frequency jamming, interferences, fading effects and multipath, is needed for WSNs with efficient resource utilization.

1.9.1 General Objectives

This thesis has three main objectives. First, a comprehensive analysis of Communication Security (COMSEC) and Transmission Security (TRANSEC) for Wireless Sensor Networks (WSNs) to find loopholes from where a malicious attack would be launched. Secondly, to modify existing COMSEC to enhance the speed of processing and reduce the amount of resources on microcontroller. Last one is the TRANSEC part which includes the design and implementation of frequency hopping mechanism and its synchronization to achieve hop rate of 100 hops/sec. for WSNs on low cost microcontroller.

1.10 Structure

A comprehensive analysis of COMSEC and TRANSEC for WSNs has been carried out in chapter 2. Chapter 3 describes testbed for implementation of proposed COMSEC and TRANSEC algorithm. First part of chapter 4 describes the modifications and enhancements made in COMSEC. Second part of chapter 3 presents the research carried on TRANSEC. Results and analysis of this research are described in Chapter 5. Chapter 6 concludes the thesis and gives some future work directions.

Literature Review

Since last decade, WSNs have been included in large number of applications. Human need of working in risky and unreachable location, attracted the researchers to solve problems at various layers. Difficulty of repairing components of WSNs is high and as they operate in unfriendly environment, nodes are more tend to fail. So low cost replaceable devices can solve problem of regular failure. Moreover, a lot of sensing nodes are used to get information of different types, so interference is occurred which causes the reduction in communication quality.

This chapter discusses literature review related to proposed system. Section 2.1 briefly describes WSNs, its applications and types. In section 2.2, sensor nodes are described. In section 2.3, a comprehensive analysis of security of WSNs is made. In Section 2.4, related work on COMSEC of WSNs is presented. In section 2.5, TRANSEC of WSNs is discussed. Section 2.6 describes related work done in TRANSEC of WSNs. In section 2.7 we discuss GFSK Modulation which is used in proposed system.

2.1 Overview of Wireless Sensor Network

In dynamic environment, physical states are sensed by using tiny sensor nodes [13]. Temperature, motion, pressure and humidity are some examples of these states. There are two types of deployments in WSNs, structured and unstructured. Unstructured deployments include operation in ad-hoc manner. Connectivity, failure detection and network maintenance are main problems of this type [14]. This problem increases as number of nodes is increased.

Whereas, in structured deployment, small number of nodes are deployed with proper planning which causes to maintain the network easily. Connectivity of nodes is managed properly and hence failure detection is easy. Main components of network include, base station, sensor nodes and field sensors [14].

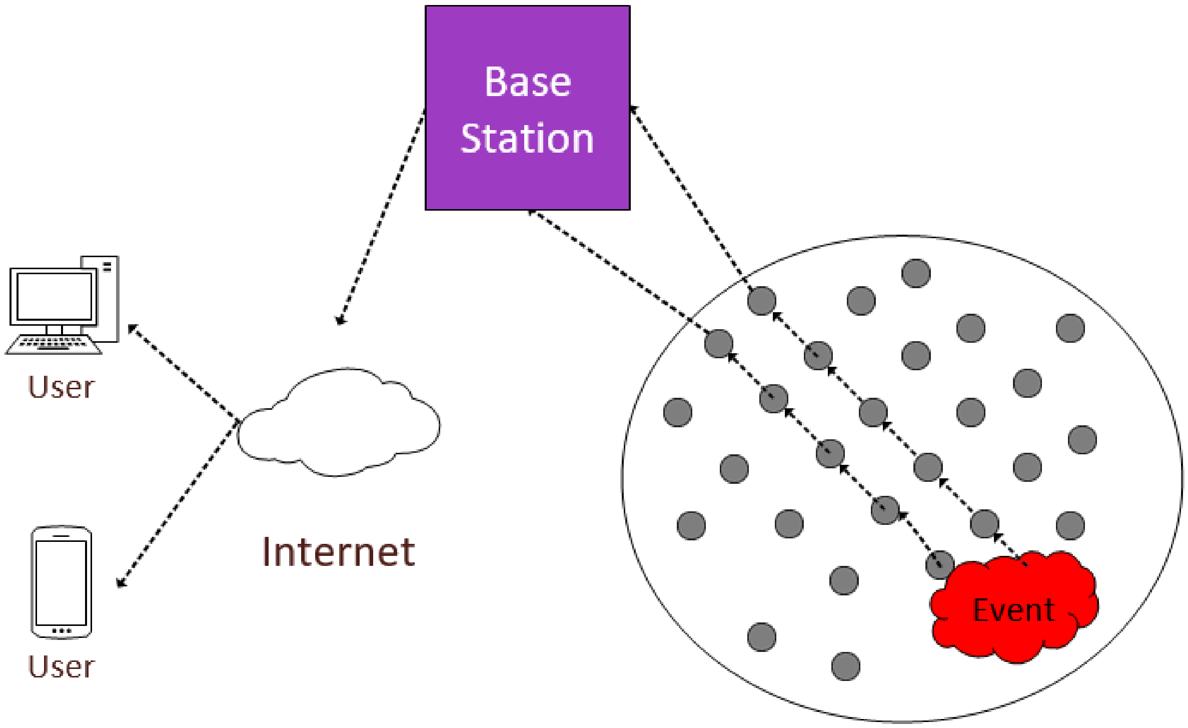


Figure 2.1: WSN Architecture

Events are monitored by nodes and they sense the surroundings and data of events is generated by their application [15]. This data is send to user devices of interest after proper processing. User devices include phones or PCs registered to networks. Decisions are made in light of collected data. Hence data security is also important. Sensor nodes are known as source nodes and base stations are known as sink nodes.

Multi hope communication is possible in WSNs as there is no capability of long haul transmission in nodes. These networks are divided into three main categories query based, event driven and periodic [16]. In query based, data is sent to sink only when a query is made by sink to get data. In event driven, nodes wake up to send data on the occurrence of particular event. Whereas in periodic, a fixed interval is defined for the transmission of data.

2.2 Hardware Modules of a Node

Sensor node has four main components [17]: a transceiver, sensor, processor and power unit. Except these, node may have, accelerometer, analog to digital converter, GPS and storage module. Figure 2.2 shows hardware architecture of sensor node.

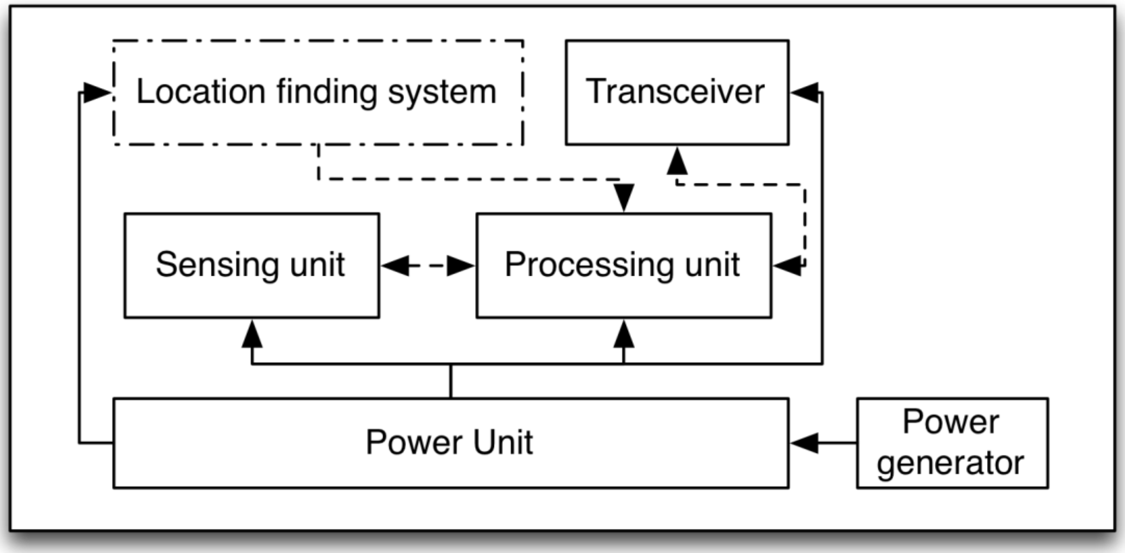


Figure 2.2: Hardware Architecture of a WSN Node

We can place multiple sensors on a node according to the requirements. Physical parameters of surroundings are sensed by the sensors. This data is converted to digital signals from analog signals. For this purpose, analog to digital converter (ADC) is used and it also give direction toward processing unit. Data is processed in processing unit and then forwarded towards transceiver to send it to its destination. Signals can be transmitted and received at same time by using these transceivers. Energy of these nodes is managed by power unit. Several algorithms are used by GPS module to find location of neighboring nodes. Data is stored in storage devices attached to nodes for further processing.

2.3 COMSEC in WSNs

The network service integrity, availability of data, authentication and network security of WSNs can be attacked [18]. All layers of protocol stack are vulnerable to attacks [19]. A simple communication model is given in Figure 2.3.

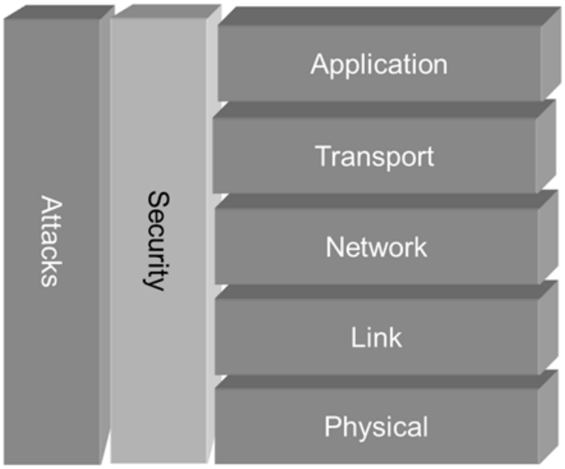


Figure 2.3: Exposed Communication Layers.

Physical layer attacks includes the extraction of sensitive data by making use of node tempering. These attacks also includes frequency jamming by means network functionality is disabled. If jamming attacks are launched by strong RF source, they are is difficult to avoid [20]. Network resilience can be improved by using spread spectrum techniques [21], but intensive computational resources are required which are not available in WSNs. If a node is attacked physically, disclosure of data is prevented by using Anti-tampering techniques [22, 23]. Malicious software can be installed on nodes which are physically accessed [24]. Usually exhaustive attacks are prevented by using standard cryptographic algorithms but their software and hardware implementations have vulnerability to fault injection and power analysis attacks. Barenghi et al. described fault injection attacks in [24]. Various levels of granularity for example, examining the inverse operation, of Rijndael cipher on the basis of side channel attacks are described by Karri et al. in [25]. Inverse relation of encryption and decryption of Rijndael is exploited in this approach and error detection mechanism is developed which analyses tradeoff between latency of error detection and area overhead. [26]

reported active and passive combined attacks. Amiel et al. proved that exponentiation of RSA can be attacked which was resistive against power analysis based attacks by making use of balanced algorithm. Injected fault must be detected immediately in order to avoid these attacks rather to detect after the computation of signature. When attack is detected, process should be aborted so that secret key can be avoided to be disclosed.

Data collision attacks are launched on link layer. Rejection of packets at reception is achieved by flipping the bits by data collisions. If these collision are repeated over times, they cause unfairness and exhaustion [27]. Error correcting code are used to handle packet collisions [28]. To prevent exhaustion attempts, rate limits in Media Access Control protocol should be applied. It will reduce these attacks up to lesser extent. Another approach is to use small frames so that channel should be captured for small amount of time.

Network layer is also exposed to different type of malicious attacks [29]. Replaying, altering and data spoofing are major attacks of network layer. Network traffic is disturbed by employing end to end latency, shortening or extending of routing paths, by false error message generation and by making routing loops. Message authentication codes are introduced to countermeasure against altering and spoofing of data. Moreover sinkholes can be created by attacker in network topology by injecting malicious nodes. Routing algorithm make it attractive to accept these neighboring nodes. Thus adversary in center diverts the traffic towards metaphorical sinkholes. This attacker created two nodes with lower latency link to base. Nodes which are located away from base station send their data towards these wormholes and hence the data of those distant nodes is tunneled towards malicious nodes injected by adversary. Commonly, wormhole attacks are combined with Sybil attacks [30]. In these attacks, multiple identities are provided by nodes to the networks and malicious nodes drop the messages and refuse to forward them. Attack named as HELLO flooding is launched on the network which make use of HELLO packet to confirm the presence of nodes in the network [31]. Protocols which depends on sending the link layer acknowledgments to check link quality are targeted by acknowledgment spoofing attacks. By making use of these acknowledgment spoofing attacks, adversary can make thing the network strong nodes as weak nodes and dead nodes as alive nodes. Authentication, link layer encryption and identity verification can resist the attacks made on network layer. Sometimes wormholes attacks are difficult to avoid. Geographic information can make protocols better against these attacks [32].

Transport layer attacks includes desynchronization and flooding attacks. Flooding attacks includes sending of lot of connections to victim nodes. Resources are allocated by victim to alive the connection. Which leads to exhaustion of memory of victim. Puzzle solving algorithm and limiting the number of connections can resist these type of attacks [33]. Desynchronization breaks the existing connection of nodes. Packet authentication can help to resist Desynchronization attacks.

Application layer attacks try to overwhelm the network by increased number of stimuli [34]. It also includes the injection of replayed or spurious packet at leaf nodes [35]. Purpose of these attacks is to waste the resources of network. Those attacks can be prevented by data authentication. A lot of solutions have been proposed to resist security threats in WSNs. Optimization of existing security algorithms is performed by using software based solutions. Insufficient processing hardware is the key limitation of these software based solutions [36-38].

The main focus of this research is application layer and physical layer security.

2.4 Related Work on COMSEC in WSNs

A secure Energy Efficient Node Disjoint Multipath Routing Protocol (EENDMRP) is proposed in [39] which detects multiple paths between destination and source by making use of energy consumption rate. [39] addresses various COMSEC threats including Sybil attack, sinkhole attack, route information altering attack and spoofing attack. Digital signature system is used to provide security. RSA and MD5 hash is used by this system.

A multipath routing algorithm basis on randomization is proposed in [40]. This algorithm randomly computes multiple paths every time when transmission of a packet is needed. So many routes can be generated potentially for each destination and source. So to attack such type of communication, adversary must have to jam all the possible routes which is not possible in practical. [40] uses three phases to secure data which includes, normal routing, random propagation of data and secret sharing of information.

A modified version of the EENDMRP (mEENDMRP) which adds transmission range settings for better energy efficiency is proposed in [41]. Two phases are involved in this protocol: Data transmission and route construction phase. Public key exchange, distance exchange and construction of routing table is performed in Route construction phase. Multiple paths on the basis of primary path are found in Data transmission phase.

A secure cluster based multipath routing protocol (SCMRP) is proposed in [42]. SCMRP is mixture of multipath and clustered sensor networks. System throughput is increased by clustered networks by making use of data aggregation. Resilience of network is increased by multipath sensor networks. Cryptographic algorithm is used by SCMRP to provide COMSEC. It restricts selective forwarding attack, tempering route information attacks and Sybil attacks.

Probabilistic multi-path redundancy transmission (PMRT) and key mechanisms to identify wormhole attacks are proposed in [43]. Key management mechanism on the basis of ID is used to build secure link and identify wormhole attack for WSNs. This technique has lower communication overhead.

Advanced Encryption Standard (AES-256) is commonly used cryptographic algorithm to provide Communication Security. The following section gives an overview of this algorithm.

2.4.1 Advanced Encryption Standard

National Institute of Standards and Technology (NIST) started a program for creation of Advanced Encryption Standard (AES) [44]. After passing tests, made on it, algorithm proposed by Vincent Rijmen and Joan Daemen named Rijndael, was accepted as new Advanced Encryption Standard. AES operates on block of size 128 bits. It has three different key lengths. One can use either on the basis of needs [44].

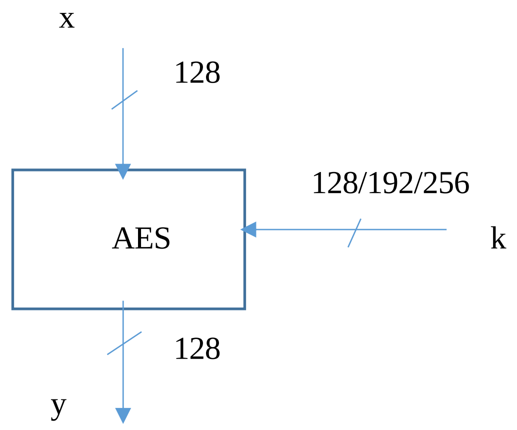


Figure 2.4: AES I/O Parameters

2.4.2 Key Length

As discussed above, three key sizes are available in AES. Each key size has different number of rounds and different mechanism to expand the key. Single round of AES encrypts 128 bits which is much larger than DES which was only 32 bits [44].

2.4.3 Internal Structure of Advanced Encryption Standard Algorithm

Internal structure of AES-256 is described in this section. Here we will examine a single round of AES-256. Data is divided into 16 bytes as A0-A16 and is fed to SBox. SBox gives B0-B16, which is permuted by shift rows and mix column layer of AES. At the end round key is added. In last round, round-key layer is avoided. AES is byte-oriented cipher [44].

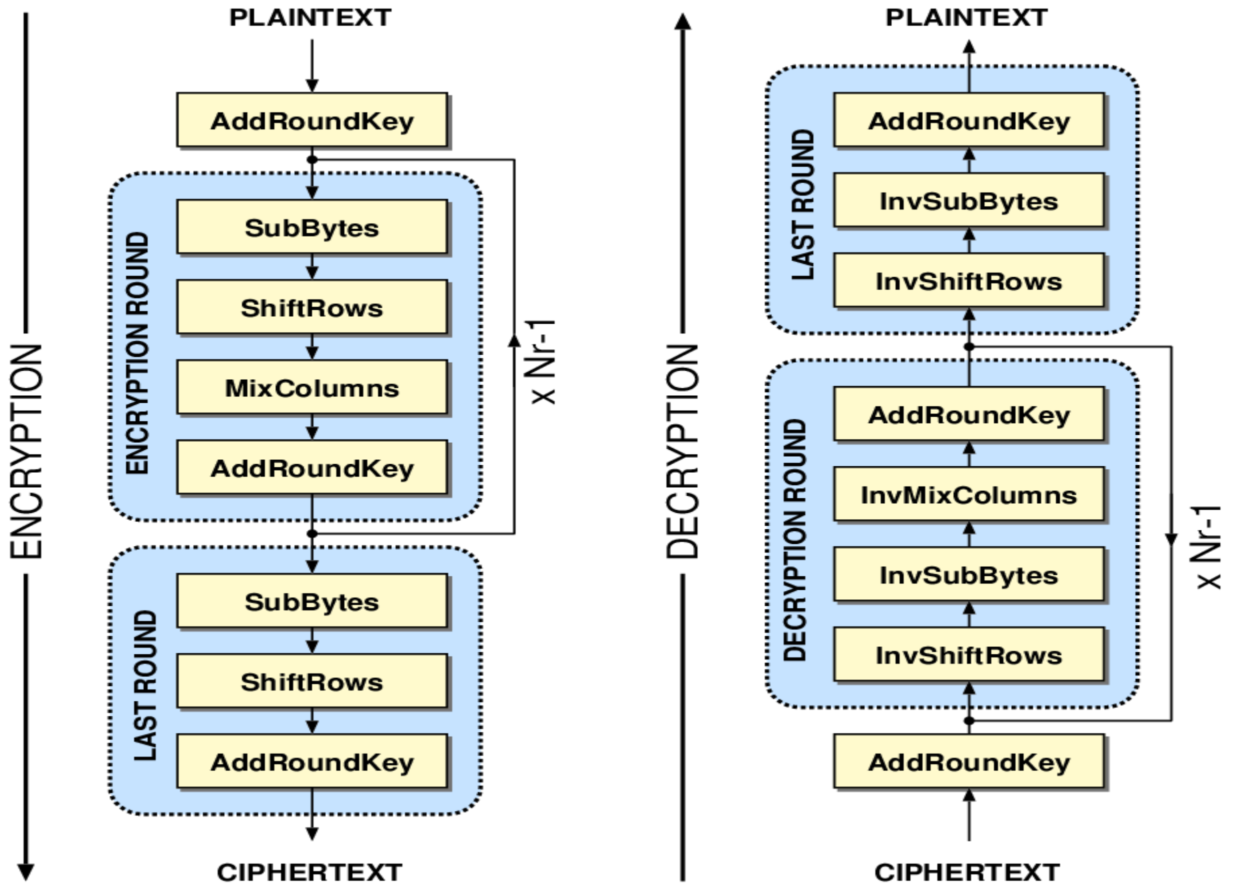


Figure 2.5: AES Round Functions

2.4.4 Overview of AES Algorithm

In initial round of AES, bitwise XOR is used to perform Add Round Key operation. In next N-1 rounds, all four steps, Byte Substitution, Shift Rows, Mix Columns and Add Round Key, are performed. While in last round Mix Columns step is skipped.

2.4.5 Key Expansion

A cipher key is used by AES Algorithm which is expended to get key schedule. Nb (Nr + 1) words are generated by Key Expansion. Nb words are initially required by the algorithm and Nb words from key are required by every Nr round. This key schedule gives linear array containing a word of 4 bytes. A function called SubWord () is used to apply the S-box on given input of 4 bytes to produce new four bytes. RotWord () function performs cyclic permutation on given word. Rcon [i], consists of round constants. Cipher Key fills first Nk words of key after expansion. Previous words XOR with earlier position of Nk word to get new word. Key Expansion for AES-256 is a bit different than other key lengths. SubWord () is applied before XOR to w[i-1] if i-4 divides Nk with no remainder.

2.4.6 Byte Substitution

To get non-linearity in cipher, S-box of 8 bits is used in AES. In Byte Substitution step, input is replaced with S-box values cross ponding to their index values. S-box must fulfil the non-linear properties to get strong cipher. Fixed points are avoided while making S-box. Figure 2.6 explains byte substitution of AES.

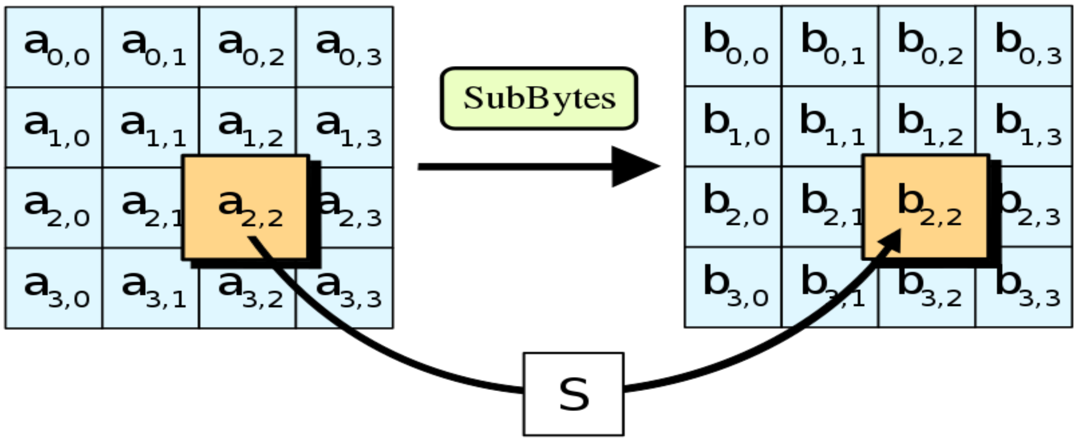

$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
S

Figure 2.6: AES Byte Substitution

2.4.7 Shift Rows

Shift Rows operation is performed on output of Byte Substitution step. This operation uses some offset to shift the bytes cyclically. In standard AES, no changes are made in first row. Left shift of one is made in second row, a left shift of two in third and finally, left shift of three is made in fourth row. Figure 2.7 shows shift rows operation in AES.

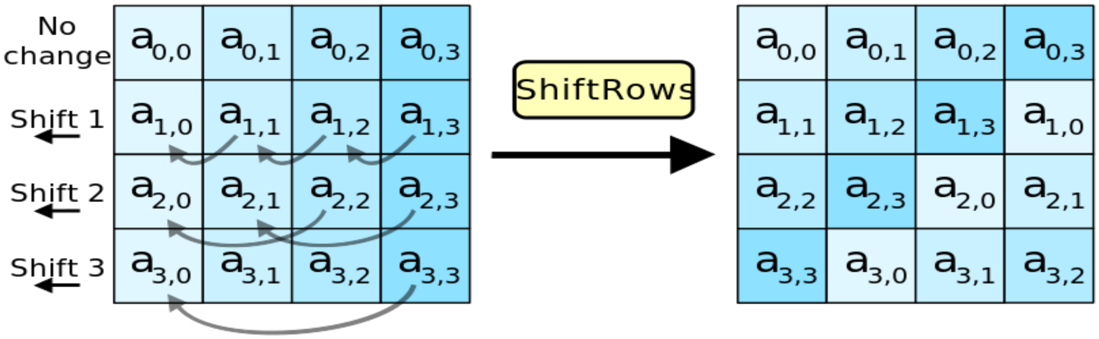

No change
Shift 1
Shift 2
Shift 3
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
ShiftRows
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,1}$ $a_{1,2}$ $a_{1,3}$ $a_{1,0}$
$a_{2,2}$ $a_{2,3}$ $a_{2,0}$ $a_{2,1}$
$a_{3,3}$ $a_{3,0}$ $a_{3,1}$ $a_{3,2}$

Figure 2.7: Shift Rows Operation in AES.

2.4.8 Mix Columns

Mix Columns step combines 4 bytes of every column of output of shift rows step by using invertible linear transformation. All outputs bytes are changed after passing through Mix Column operation. This operation is used to get diffusion in resulted cipher. In this step, when a matrix is left multiplied by column, it produces new value. Addition and multiplication of entries is carried to get multiplication of Matrix. Multiplication in AES is modulo irreducible polynomial whereas, addition is simple XOR. Figure 2.8 explains Mix Column operation of AES.

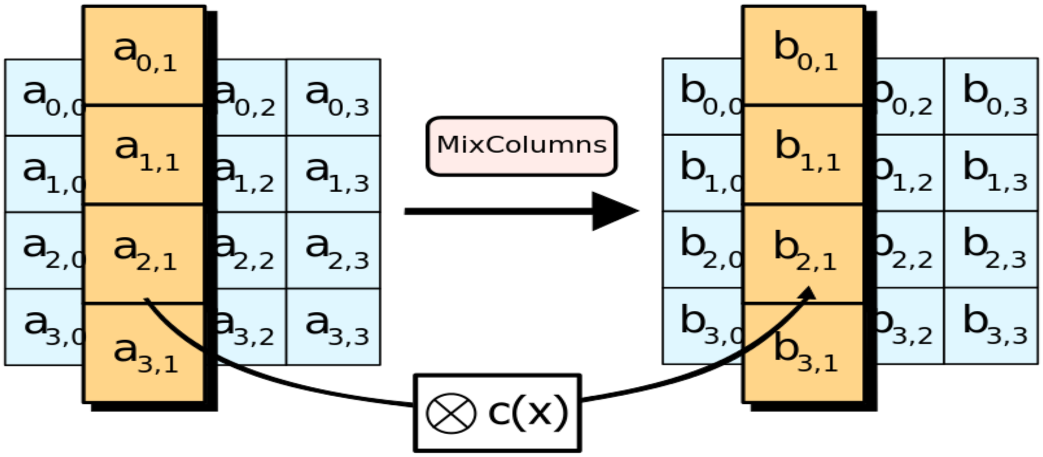
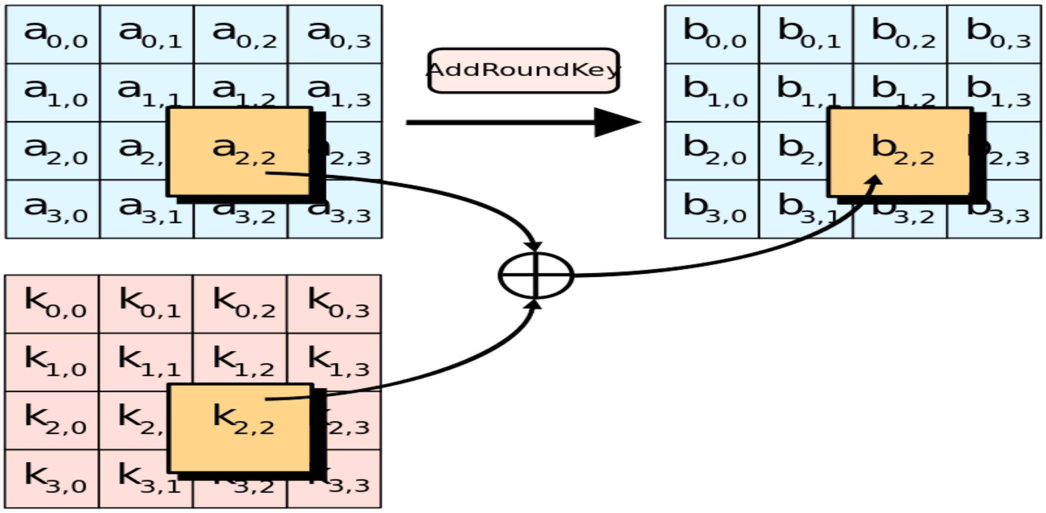

$a_{0,1}$ $a_{1,1}$ $a_{2,1}$ $a_{3,1}$
$a_{0,0}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,2}$ $a_{3,3}$
MixColumns
$\otimes c(x)$
$b_{0,1}$ $b_{1,1}$ $b_{2,1}$ $b_{3,1}$
$b_{0,0}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,2}$ $b_{3,3}$

Figure 2.8: Mix Column Operation in AES

2.4.9 Add Round Key

In Add Round Key operation, subkey which is calculated using key expansion of AES, is combined with output of Mix Column operation. Size of state and subkey is same. Bitwise XOR is used for combining state with subkey. Figure 2.9 explains operation of Add Round Key in AES.



AddRoundKey
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
$k_{0,0}$ $k_{0,1}$ $k_{0,2}$ $k_{0,3}$
$k_{1,0}$ $k_{1,1}$ $k_{1,2}$ $k_{1,3}$
$k_{2,0}$ $k_{2,1}$ $k_{2,2}$ $k_{2,3}$
$k_{3,0}$ $k_{3,1}$ $k_{3,2}$ $k_{3,3}$
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$

Figure 2.9: Add Round Key Operation in AES

2.4.10 Decryption of Advanced Encryption Standard

For decryption of Advanced Encryption Standard (AES), each layer is inverted to get decrypted output. Key expansion is made in similar way for decryption also. In first round, mix column layer is skipped, as it was skipped in last round of encryption [44].

2.4.11 Inverse Shift Rows

Inverse shift Rows is inverse of shift Rows operation. Different numbers of offsets are used to cyclically shift the three lower rows of cipher. Offset for shift is calculated as Nb-shift.

2.4.12 Inverse Sub Bytes

Inverse of Sub Bytes is known as Inverse Sub Bytes operation. In this step, state is replaced by inverse S-box values. Affine transform along with multiplicative inverse is used to calculate inverse S-box.

2.4.13 Inverse Mix Column

Inverse of Mix Columns operation is called Inverse Mix Columns transform. Each column is treated as four term polynomial and this operation is performed on state column by column. All operations are performed in GF (2^8) field.

2.4.14 Inverse Add Round Key

Inverse of Add Round Key is known as inverse Add Round Key. In inverse add round key, XOR operation is performed on subkey and state to decrypt the cipher.

2.5 TRANSEC in WSNs

In order to ensure communication quality of WSNs, link quality plays a vital role. For WSNs, above 90% data reliability is required in worse environment to monitor the data. As WSNs has their wider use in ISM band which has greater amount of attenuation and multipath interference, link quality is affected seriously. To improve quality of communication in such a harsh environment, Frequency hopping spread spectrum technique is of great importance.

In standard Frequency Hopping Spread Spectrum (FHSS) systems, receiver and transmitter repeated hope based on pseudo random sequence which is known to both of them. Carrier frequency is selected based on this sequence in order to transmit data. Dwell time is the time interval used on each channel by transmitter. Channel bandwidth (Δf) of each hopping channel is same as that required for application. System processing gain (Gp) can be defined as follows [45], [46]:

$$Gp = \frac{\text{RF Bandwidth}}{\text{Message Bandwidth}} = \frac{M\Delta f}{\Delta f} = M \qquad (2.1)$$

Main benefit of FHSS, as opposed to fixed channel communication, lies in equation (2.1). In fixed channel communication, performance of system is decreased due to constantly occupying of bandwidth at same frequency for long time. So FHSS systems have better performance in narrow bands as they allocate RF emission by using a series of M channels which provide net total hoping bandwidth (Wss).

On low power SoCs, Super Response (SR) FHSS is used. Spreading of sequence is not transmitted in this scheme, so synchronization between receiver and transmitter in frequency and time domain is required. FHSS scheme can be fast or slow according to the requirements [47-50]. Receiver should be able to achieve synchronization by analyzing the signal received from transmitter [49], [51].

2.6 Related Work on TRANSEC in WSNs

A link evaluation algorithm, based on average RSSI (Received Signal Strength Indication) and mean LQI (Link Quality Indication), is proposed in [52]. This algorithm depends on EPA (Ethernet for Plant Automation) wireless technology. Drawback of this algorithm is that hardware with greater resources is required that must support RSSI and LQI. Another technique depending on prediction of channel is proposed in [53]. This technique makes use of channel prediction techniques in order to enhance model of prediction of channel. To improve performance of FHSS, [54] describes estimation of threshold of RSSI of channel. For WSNs, adaptive frequency hopping algorithm is proposed in [55]. This algorithm make use of Mesh network. This implementation requires network topology and hence for all networks, it is not applicable.

FHSS helps to provide transmission security and to avoid interference by handling the transitions of carrier frequencies of the system [56]. Figure explains peer to peer FH system whose transmitter and receiver make use of different transmission channels. When transmission is started, one channel is used until interference is observed. As soon as interference is monitored, they jump to other known channel. It is proved with research that network reliability can be improved by using frequency hopping techniques [57].

2.6.1 Channel Assessment

Channel assessment plays an important role in WSNs to make decision for frequency. A few years ago, there was no good method for measuring the channel [58]. Researchers assume wireless channel symmetrical and ideal. But studies proved that asymmetric properties are present in wireless channel. Communication distance, obstacles and interference affects the channel quality. So there are a lot of factors which restrict to assess channel quality. Some people proposed serial numbering of packets to find loss packet rate as it reflects the status of link quality. But control packages are required to accomplish this strategy [59]. It has less link status sensitivity. Wireless chips such as CC2530 and CC1100 offers better sensitivity and effectiveness in measuring the quality of link [60]. RSSI value is improved by inquiring values of register of RF modules. No network overhead is imposed on network as packet sends RSSI values. Hence RSSI is used to measure the assessment of channel.

2.6.2 Conditions on RSSI

In WSNs, frequency hopping decision cannot rely on RSSI value as quality of channel is not only changed by interference hence signal interference does not rely on RSSI directly:

1) Environment factor plays an important role in changing Received Signal Strength Indication value. RSSI values are affected by change in distance, line of sight, multipath effect and path reflection.

2) RSSI value is changed if a signal of same frequency also entered the same channel which changes RSSI value.

Hence in constantly changing environments, RSSI is no more effective measure for frequency hopping. As we cannot determine that either RSSI is changed by interference or environment.

2.6.3 Synchronization for FHSS

Frequency and time are the domains of uncertainty while synchronization between receiver and transmitter is considered. At receiver, two processes are carried to get phase synchronized local code, which are known as tracking and acquisition. When synchronization is lost, due to noisy channel, acquisition takes place.

Synchronization is provided by acquisition by limiting the values to finite numbers of frequency and timing and frequency offsets [49]. In acquisition state, acquisition time is taken by system and it depends on algorithm used and search scheme. Acquisition techniques are based on parallel or serial search approach which are known as matched filter techniques. Correlation process of these two mechanisms is same, which gives an idea that on reception, how much similarity be there for synchronization. As the phase of sequence which is locally generated is brought, acquisition process is handled by control sub-system at receiver [49]. Tracking system is activated after above condition is verified and detected. More detail on serial and parallel acquisition are available [46], [49], and [61].

Parallel search techniques has fastest time of acquisition due to simultaneous examination of possible code offsets. But these are expensive to implement due to same number of matched filters as that of hopping carriers. Serial search is used more commonly. In this engine, alignment cells or trails are continuously performed [49]. Cell is similar to uncertainty domains which are discussed earlier, frequency and time. Actual cell is not accepted if test results of certain test on cell are not good. If rejection of cell is occurred, search is made again by changing phase of sequence. If this is not the case, acquisition is declared and phase of tracking is triggered.

In [62], another method for synchronization is described, known as Time of Day (TOD). Information of synchronization is stored in variable named TOD which is placed in header of synchronization. Specific frequency of synchronization is used to transmit this header. Then synchronization information is extracted by receiver from header of synchronization and local generator of frequencies uses this information to generate frequency hopping sequence and then synchronization is achieved by receiver. Only synchronous recognition is required by receiver instead of data.

Drawback of this technique is that slighter change in time of transmitter and receiver, cause out of synchronization of transmitter and receiver.

For wireless sensor networks, above mentioned techniques for synchronization are impractical due to their computation complexity because low cost microcontrollers are commonly used in WSNs.

2.6.4 Randomness of Hopping Sequence

When huge number of nodes are connected in WSNs, chances of collision of frequencies are increased significantly. Which degrades the performance of communication [63], [64]. Many Frequency hopping techniques focus to reduce the cost and complexity of nodes and to make them power efficient [65], [66].

Randomness for hopping is guaranteed by using output of pseudo random number generator (PRNG) output which is based on some seed. Output of PRNG is used as pointer to look up table for frequencies [67]. If large number of frequencies are used for hopping as hard codded table, ROM limitation does not allow this to happen [67]. This causes trade off with respect to performance of system as it is related to number of hope frequencies. Less memory consuming algorithm can be used which could calculate table at runtime to avoid this problem [67].

Performance of FHSS system highly depends on randomness of hopping sequence. One can get more details on generation of pseudo random generation in [50], [59], and [68]. Cross correlation properties are also of great interest. If sequence has good cross correlation properties, less interference will occur which will cause high throughput of the system. However in practice, it is difficult to achieve to imperfection of system.

In the light of given literature, following two questions are present in frequency hopping spread spectrum algorithms:

1) Hardware resources for many frequency hopping algorithms are very high.

2) Some algorithms require special kind of network topology. Hence universal applicability is lacked in these algorithms.

This thesis presents a frequency hopping algorithm along with synchronization technique and efficient implementation of Advanced Encryption Standard (AES-256), for multi-channel RF module NRF24L01+, used in WSNs

For WSNs, if the system is hopped wrongly, it will have great impact on communication. Hop decisions for FH strategy will become difficult to handle by the system. In addition, burden of communication in network will be increased by wrong hopping decisions which will reduce stability of the system. So, decisions for frequency hopping must be able to meet the requirement with less hardware consumption is the purpose of this research.

2.7 GFSK Modulation

Gaussian Frequency Shift (GFSK) is a form of Continuous Phase Frequency Shift Keying (CP-FSK), which is further a modification of Discontinuous Phase Frequency Shift Keying (DP-FSK).

2.7.1 Gaussian Frequency Shift Keying

GFSK can be observed as form of CPFSK modulation. High frequency components, which are present in output spectrum of modulated signal, are minimized in CPFSK due to constantly changing of phase of CPFSK modulated signal. In GFSK, Pulse Shaping Gaussian Low Pass Filter is used to pass baseband signal prior to modulate it which will give shape of half-sinusoidal to pulses to make phase trajectory of FSK signal smoother. This modulation technique has following benefits:

In GFSK, modulated signal has constant envelope. So, signal can be operated without Spectrum Regeneration by using a Class-C Power Amplifier. Hence high power efficiency is increased. Secondly, in GFSK, spectral Side Lobes are in lower level as compared to DPFSK. Due to this, interference from nearby channels is minimized due to which, spectral efficiency is increased. This is particularly important in case of non-linear channel. Non-Coherent Demodulation schemes can be used to demodulate GFSK modulated signals due to which cost of GFSK receivers is decreased. [60][61].

Spectrum of the FSK modulated signal is made narrower by using pre modulation low-pass filtering. Time domain response of low-pass filter is well behaved and hence is called Gaussian filter. Gaussian Low pass Filter has Gaussian frequency response and is given by:

$$H(\omega) = \tau\sqrt{2\pi}e^{-\frac{(\tau\omega)^2}{2}} \qquad (2.2)$$

here: ω = frequency (radians/sec)

τ = constant

This represents shape of the Normal Probability Density Function or Gaussian.

Impulse response of filter which has a Gaussian frequency response is also Gaussian. It is verified by inverse Fourier transform of the frequency response.

$$h(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(\omega)e^{-j\omega t}\,d\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty} \tau\sqrt{2\pi}e^{-\frac{(\tau\omega)^2}{2}}e^{-j\omega t}\,d\omega \qquad (2.3)$$

which gives

$$h(t) = e^{-\frac{1}{2}(\frac{t}{\tau})^2} \qquad (2.4)$$

Gaussian filter has well behaved impulse response with no overshoot. Whereas, frequency response of this filter falls off slowly. In general, time domain response is better if frequency domain response is more gradual. Among all analog filter types, Gaussian filters have highest gradual frequency domain responses [62] [63].

Gaussian Filter has linear frequency domain response hence phase of Gaussian filter is also linear. The bandwidth of the Gaussian Filter is given by

$$\mathrm{T} = T_b = 1/f_b \qquad (2.5)$$

here: T or T_b = Duration of the filter or Bit Period

f_b = Bit Rate

If B is 3-dB bandwidth, then response can be treated as Relative Bandwidth

$$\mathrm{BT = Filter\ Bandwidth.\ Bit\ Period} = \frac{FilterBandwodth}{Bitrate} \qquad (2.6)$$

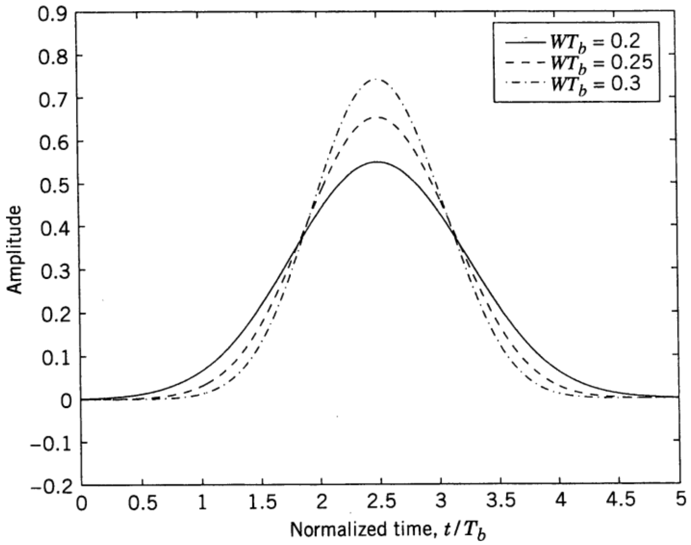


Figure 2.10: Gaussian Filter Bandwidth Effect on Signal Frequency-Spectrum

GFSK modulated signal with infinite bandwidth can be given as:

$$\text{S(t)} = \cos[2\pi(f_c + (I_m D[k]P[t - nT]/2T_b)t] \qquad (2.7)$$

here:

$\sum_{n=0}^{N} D[k]P[t - nT_b]$ = Binary Data in form of Rectangular Pulse Stream.

$1/\,T_b$ = Baseband Modulating Signal Bit Duration

f_c = Carrier Frequency

I_m = Modulation Index

Transfer Function of Gaussian Filter is given by:

$$G(\omega) = A_0 e^{-\alpha\omega^2} e^{-j\omega t_0} \qquad (2.8)$$

here:

$$\alpha = (ln2)/2{B_{3db}}^2)$$

Gaussian Filter's impulse Response is given by:

$$H(t) = (A_0/\sqrt{\pi})\beta\, e^{-[(t-t_0)\beta]^2} \qquad (2.9)$$

here:

$$\beta = \sqrt{\frac{2}{ln2}\pi f_c B T_b}$$

Hence GFSK modulated signal is given by:

$$S(t) = A_0 \cos[2\pi f_c t + (2\pi I_m/2T_b) \int_0^t ((D(\tau)\otimes \mathrm{H}(\tau))/2T_b d\tau] \qquad (2.10)$$

Chapter 3

The Prototype Test Bed

3.1 Overview

A simple test bed for implementing COMSEC and TRANSEC, is combination of a DHT11 a temperature and humidity sensor at transmitter side, 2xNRF24L01+, 2xATmega328, 2xPower Supply and a LCD connected to the receiver side.

3.2 Nordic nRF24L01+

The Nordic nRF24L01+ transceiver IC is used in proposed work. It operates in Industrial, Scientific and Medical (ISM) band of 2.4 GHz. It has maximum range of 1 Km by using external antenna and a capacitor. This transceiver operates at low voltage ranges from 1.9-3.6V and maximum current used by the module is less than 14 mA. Hence it provides ultra-low power solution in wireless sensor networks. Wireless connectivity of nRF24L01+ with microcontrollers is very robust. This transceiver makes use of Gaussian Frequency Shift Keying (GFSK) modulation. It can be connected with MCU by using Serial Peripheral Interface (SPI) [69, 70]. This transceiver is shown in figure 3.1.

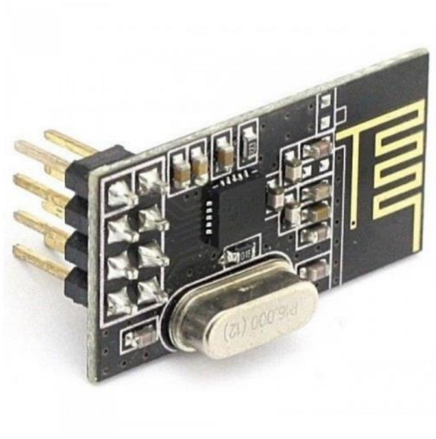


Figure 3.1: Nordic nRF24L01+

The Block diagram of nRF24L01+ is shown in Figure 3.2. Operation modes of nRF24L01+ are TX, RX, Power Down and standby. This module provides 126 channels for RF communication and they ranges from 2400-2525 MHz. Bandwidth of each channel is 1 MHz when data ate is set up to 1 Mbps. [70, 71].

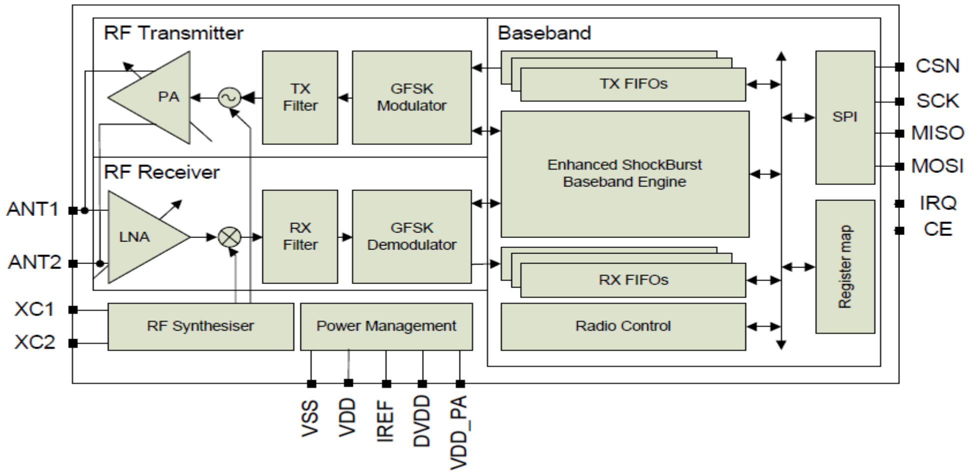


Figure 3.2: Block Diagram of NRF24L01+ Transceiver

3.2.1 TX mode

When nRF24L01+ operates as receiver, active mode is RX mode. PRIM_RX, PWR_UP and CE bits must be set to high in order to enter this mode. In this mode, received signal is demodulated using GFSK demodulation. This demodulated data is sent to baseband protocol engine, which is constantly searching for valid packets. If valid packet is arrived, payload is set to vacant. Received packet is discarded if RX FIFO is not empty. Carrier detect is present in RX mode. When a FSK modulated signal is arrived, it is detected.

3.2.2 RX mode

When nRF24L01+ operates as receiver, active mode is RX mode. PRIM_RX, PWR_UP and CE bits must be set to high in order to enter this mode. In this mode, received signal is demodulated using GFSK demodulation. This demodulated data is sent to baseband protocol engine, which is constantly searching for valid packets. If valid packet is arrived, payload is set to vacant. Received packet is discarded if RX FIFO is not empty. Carrier detect is present in RX mode. When a FSK modulated signal is arrived, it is detected.

3.2.3 Data Rate

Maximum data rate of nRF24L01+ is 2 Mbps but receiver sensitivity is better up to 3 dB when it operates at 1 Mbps. Average current usage is less at high data rates due to less chances of collisions. Receiver and transmitter must be at same data rate to make communication possible.

RF_DR bit is responsible to set data rate and this bit lies in RF_SETUP register.

3.2.4 LNA Gain

Low Noise Amplifier (LNA) gain setting controls the gain at receiver side. It can be set by using LNA_HCURR bit of RF_SETUP register. 0.8 mA of current can be saved by reducing 1.5 dB receiver sensitivity.

3.3 Microcontroller

Microcontroller board Arduino Uno has 14 I/O pins, 16 MHz crystal oscillator, 6 analog inputs, power jack, rest button and a USB connection [44]. It is based on ATmega328. Arduino is open source environment which is used to write software for board. One can use sensors, switches, motors and other physical states to develop interaction with objects.

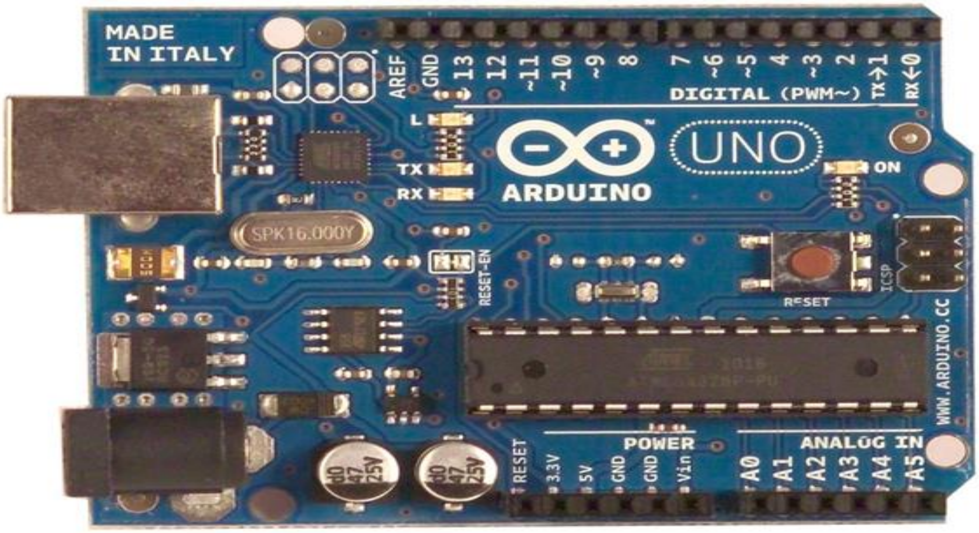


Figure 3.3: ATmega328

Arduino makes it easy to work with microcontroller and it has following features.

3.3.1 Features of Arduino Uno

Arduino Uno is cross platform supported. It can run on Windows, Linux and Mac OS. These boards are inexpensive as compared to other MCUs. Arduino environment is easy to use for people with no experienced and is highly flexible. Software of Arduino is open source and C++ libraries can be used as extensions for ease of use. This board is inherited from ATMEGA168 and ATMEGA8 microcontroller by Atmel. So modified version can be made with ease by experienced developers.

3.3.2 Specification

Clock speed of ATmega328 is 16 MHz and it has 1 KB of EEPROM. Static Random Access Memory (SRAM) of this microprocessor 2 KB. It has flash memory of 32 KB from which 0.5 KB is used by bootloader. Every I/O pin consumes a DC current of 50 mA for 3.3V pin and 40 mA for 5V pin. It contains 6 analog pins and 14 digital input/output pins.

3.3.3 Power

Input voltage of 5V is recommended to operate ATmega328. Power to this microprocessor can be given by a 5V DC battery or USB cable. We can also connect a DC adapter to power up the module.

3.3.4 Communication

Arduino Mega328 can communicate with other microcontrollers or computers by using UARTs at 5V for TTL serial communication. IDE of Arduino includes serial monitor to send and receive text data from and to the board. When data is being received or transmitted, TX and RX LEDs of board began to flash. SPI and TWI communication is also supported by ATmega328. Libraries can be used to make SPI and TWI communication.

3.3.5 Programming

Programmer is embedded on the board and it uses STK500 protocol to communicate. A bootloader, to burn new codes, is pre burned on ATmega328. New programs can be uploaded using Arduino IDE.

3.4 DHT11

DHT11 senses the humidity and temperature by using calibrated digital signal. Technology of DHT11 make sure that it is stable for long time and is reliable. It has NTC temperature device and resistive element. This sensor has anti-interference ability, quick response and low cost. Humidity chamber of this sensor is very accurate and is stored in OTP memory. Signals in process are detected in internal sensors. It can sense signals from 20 meters which makes it useful in various applications. This sensor has four pins in a single row.

3.5 LCD 16x2

A 16x2 LCD is used to observe out of the system. This LCD has 16 pins and operates under voltage of 5V. If we see from left to right, 1st pin is ground, 2nd Vcc, 3rd is Vo pin. Potentiometer is attached to this pin to control contrast. 4th is register select or RS pin to send data or commands towards LCD. 5th is read write R/W pin which determines whether we want to read or write to LCD. 6th pin is named E and writing to registers is enabled by this pin. Next D0-D7 are data pins to send data of 8 bits. Anode “A” and Cathode “K” pins of LCD are used to provide backlight.

At transmitter side Temperature and Humidity sensor is connected to Arduino Uno to collect data from environment. After collection of data, processing is performed in microcontroller. Then RF module is there to perform transmission of data. At receiver side RF end receives data and sends it to MCU for processing and then output is shown on LCD connected with microprocessor.

3.6 GNU Radio

To capture spectrum of transmitted signal, Software Defined Radio techniques are used. GNU Radio is open source platform for Linux OS. The GNU Radio is used as framework for building and running signal processing applications or software based radio. In GNU Radio, signal processing blocks are connected in series to form an executable flow-graph. A single general purpose front end can be used instead of using different designs, by making use of GNU Radio. Processing blocks can be created in python or C++ or a mixture of C++ and python. Swig module will behave as a wrapper while using mixed languages.

3.7 Universal Serial Radio Peripheral 2 (USRP2)

The Universal Serial Radio Peripheral (USRP) is a Software Defined Radio Device and commonly used as RF front end in SDR applications. It offers a bandwidth of 56 MHz and provides frequency range up-to 6 GHz. For synchronization and timing precision, GPS-disciplined oscillator (GPSDO) is present in USRP. Wireless applications can be created using GNU Radio and USRP for example direction finding, FM radio and passive radar.

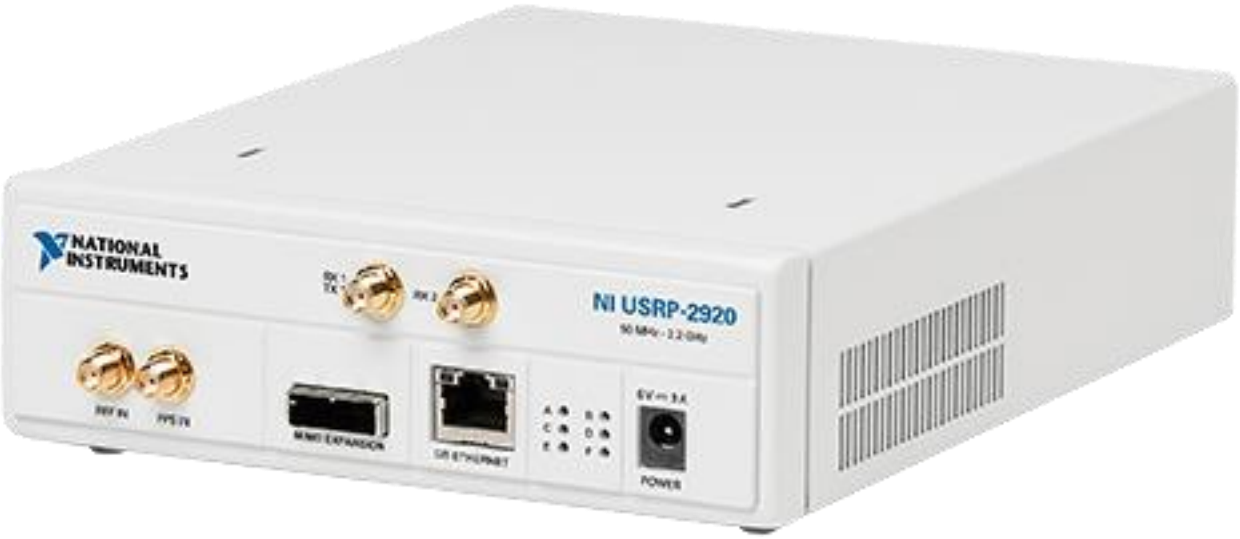


Figure 3.4: NI USRP-2920

3.7.1 Design

Varity of models of USRP are available and their architecture is same. Motherboard of USRP provides: power regulation, host processor interface, DACs, ADCs, FPGA, synchronization and clock generation. These components provides processing of signals at baseband level. Daughterboard. Which is a modular front end of USRP, provides analog operations such as filtering, up/down conversion and conditioning of signals. This modular property of USRP allows to operate between 6 GHz and DC. Basic task of FPGA is to perform DSP operations to provide translation of analog domain real signal to complex, lower rate digital domain baseband signals. FPGA code is open source and can be modified to allow low latency and high speed operation in FPGA. Block diagram of USRP devices is given in figure 3.5.

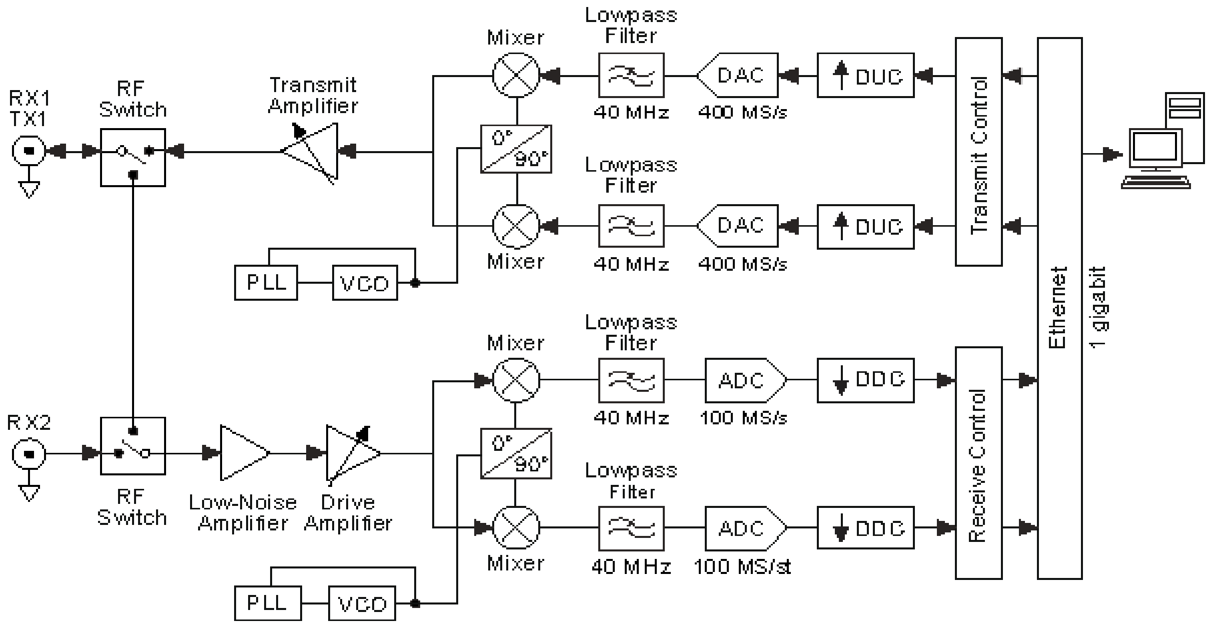


Figure 3.5: Block Diagram of USRP

3.7.2 Software

Ettus Research has provided USRP hardware driver (UHD). It supports MacOS, Linux and Windows platforms. Many frameworks including Redhawk SDR, GNU Radio, MATLAB and LabVIEW make use of UHD. UHD API can be used to directly access UHD functionality in C++. UHD can be used in any language which can import C++ functions. In python SWIG performs this functionality.

3.8 GR-Fosphor

GR-Fosphor is GNU Radio block which is used as software based real time spectrum analyzer. It is flexible spectrum analyzer which provides details of spectrum of RF signals with great precision. This GNU-Radio block make use of gr-osmosdr and supports all SDR frontend devices which are compatible with GNU-Radio. All functionalities of zooming and span settings are available in this block. We can also analyze waterfall plot of received spectrum using gr-fosphor. Gr-fosphor utilizes OpenCL and OpenGL acceleration. This tool is used to analyze the spectrum of proposed system. OpenCL (Open Computing Language) provides a framework to write programs that runs across heterogeneous systems which comprised of graphics processing units (GPUs), central processing units (CPUs), field programmable gate arrays (FPGAs), digital signal processors (DSPs) and hardware accelerators. Open Graphics Library (OpenGL) is a cross language and cross platform application programming interface (API) which renders 3D and 2D graphics vectors.

Instructions of installing and using gr-fosphor are given in Appendix-I

System Design and Implementation

4.1 Overview

First part of this chapter describes the enhancements made in Advanced Encryption Standard (AES-256). A simulator has been written in C++ to compare the results of our implementation with previously presented well known implementations. Second part of this chapter proposes a TRANSEC algorithm and its synchronization mechanism for WSNs with limited amount of computation resources.

4.2 COMSEC

Standard NIST document is followed to implement AES-256 in C++ programming language, after testing the functionality of algorithm, following changes were made to make it robust and memory efficient. After optimization, it is confirmed that no change in functionality is made. A simulator in C++ is written to verify that this modified implementation is light weight and more robust than previous implementations.

4.2.1 Lookup Table Definition and Placement

As microprocessor used in this research is ATmega328 which has only 2k bytes of RAM so lookup tables are marked constant so that they can be placed in read-only storage instead of RAM. In this platform ROM is larger than RAM so lookup tables are hard coded and are placed in ROM. Lookup tables in this implementation are defined as follows.

“static const uint8_t”

4.2.2 Key Schedule

For key schedule only required 7 values of Rcon are kept and all other unused entries are deleted to save ROM. As symmetric cryptographic encryption algorithm is used and expanded key remains same, so it is expanded during initialization instead of expending it every time. It saves processing resources by calling key expander only once.

4.2.3 Conditional Logic Optimization

During conditional statements execution, at first, condition with greater probability of false, is checked. It makes the implementation robust because resources are not wasted on checking conditions which are of no interest.

4.2.4 Variable Definition

Char and short are avoided to use as local variables especially for loops because for short and char types, at the end of each assignment, size of local variables needs to be reduced to 16 or 8 bits. This is known as sign extending for signed and zero extending for unsigned variables and is implemented using left shift of registers by 16 or 24 bits tailed by unsigned or signed shift. It takes two instructions which makes it less efficient for processing.

In this implementation, for local variables, we used unsigned int to avoid these shifts. This approach is more useful when we deal with buffers and this is the case when we have to apply encryption on a buffer of 16 bytes. So, all variables for internal loops and other processing, are defined as unsigned to increase speed and decrease code size.

4.2.5 Mix Columns

Mix column operation is the most complex operation of AES algorithm. It involves multiplications in GF (2^8) field which are difficult to perform on low resource hardware. Mix Column operation of AES shows that during encryption process, we need to multiply with 2 and 3 only. So pre calculated lookup tables to perform multiplication are placed in ROM. Hence, we don't need extensive multiplications during encryption process instead we simply use these lookup tables to perform multiplications and only simple XORs will be used. Hence no multiplication is involved in Mix Columns operation at cost of ROM which is much greater than RAM in case of ATmega328.

4.2.1 Inverse Mix Columns

For inverse Mix Columns operation, we need to perform multiplications with 9, 11, 13, and 14. Placing four lookup tables in ROM is not a good idea because they cost ROM. So a function is implemented to multiply in the field GF (2^8) to reduce the code size and increase the performance.

```
static uint8_t tm_x_fun(uint8_t a)
{
  return ((a<<1) ^ (((a>>7) & 1) * 0x1b));
}
static uint8_t Mul_fun(uint8_t x, uint8_t y)
{
  return (((y & 1) * x) ^
       ((y>>1 & 1) * tm_x_fun (x)) ^
       ((y>>2 & 1) * tm_x_fun (tm_x_fun (x))) ^
       ((y>>3 & 1) * tm_x_fun (tm_x_fun (tm_x_fun (x)))) ^
       ((y>>4 & 1) * tm_x_fun (tm_x_fun (tm_x_fun (tm_x_fun (x))))));
  }
```

4.2.2 Valgrind

Valgrind is an open source Linux based command line tool which is used to test memory error of c or c++ codes. It is powerful tool to detect memory leak. After completing the implementation, it was passed from Valgring to confirm that no memory leak is present in it.

Instructions of installing using Valgring are given in Appendix-II.

4.2.3 Simulator

A simulator in Visual Studio 2017 using C++ has been written to make a comparison of robustness of our implementation with other open source implementations of Advanced Encryption Standard (AES-256) i.e. OpenSSL and CryptoPP. This simulator make use of ctime and chrono of C++ language to detect the robustness of algorithms. All the processing of encryption algorithm is wrapped between two functions of simulator to detect the robustness. Timing values use millisecond unit of time to describe the robustness. This simulator makes use of high resolution clock to determine timing values precisely.

Source code of Simulator is given in Appendix III

4.3 TRANSEC

A frequency hopping algorithm and its synchronization mechanism is designed for multichannel WSN transceivers to implement TRANSEC. Nordic nRF24L01+ is used to verify functionality of algorithm.

4.3.1 Transmitter Design

Transmitter of proposed system is consisted of four main components. Sensing device is a sensor named DHT11, a well-known humidity and temperature sensor. This sensor gathers information from surroundings and then this data is captured in array for further processing, in microcontroller. So the main task of this part is to collect data from environment and send it to microprocessor. Overall subsystems are controlled by microprocessor and it and completes the processing of data received from the sensors in form of array. This array of data is encrypted in microprocessor using modified AES-256 encryption algorithm described above and transmitted towards the receiver side by using a multichannel WSN module by Nordic nRF24L01+. Power unit is necessary for the system to work and is included with the system as 5V recharge able DC battery.

The block diagram of the proposed transmitter system is shown in Figure 4.1.

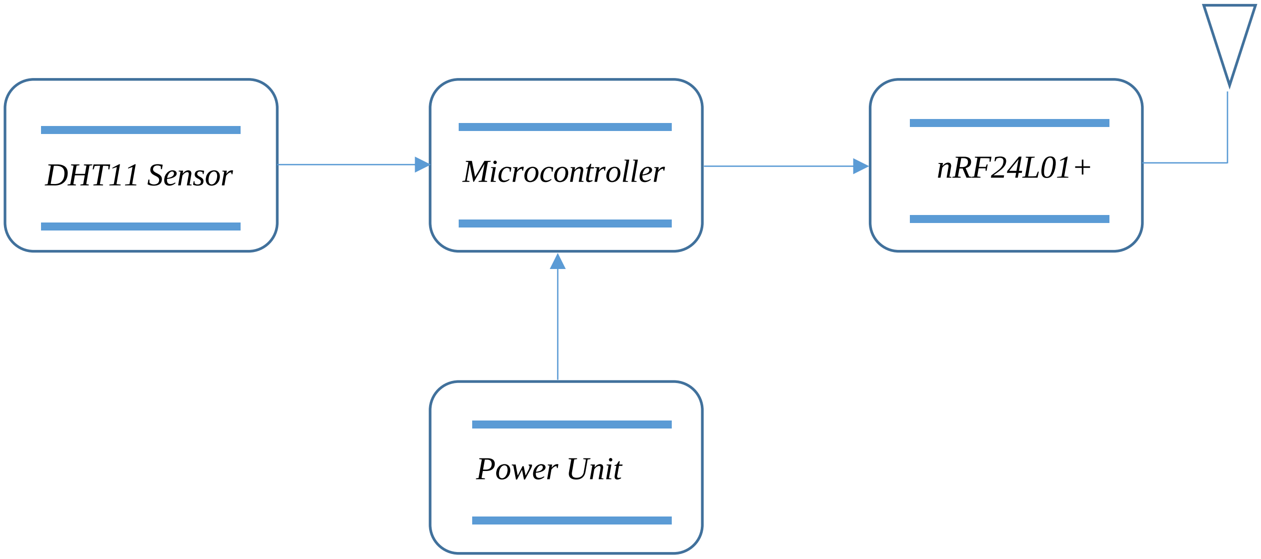


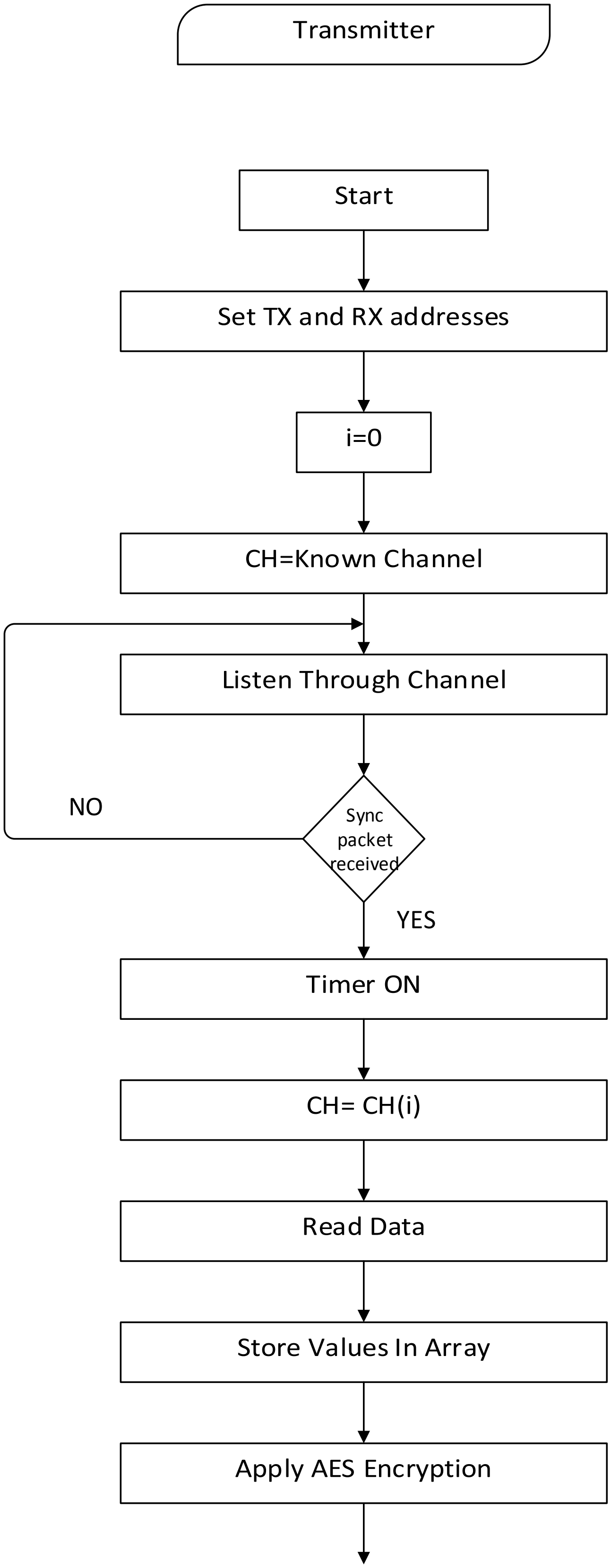
Figure 4.1: Transmitter Subsystem

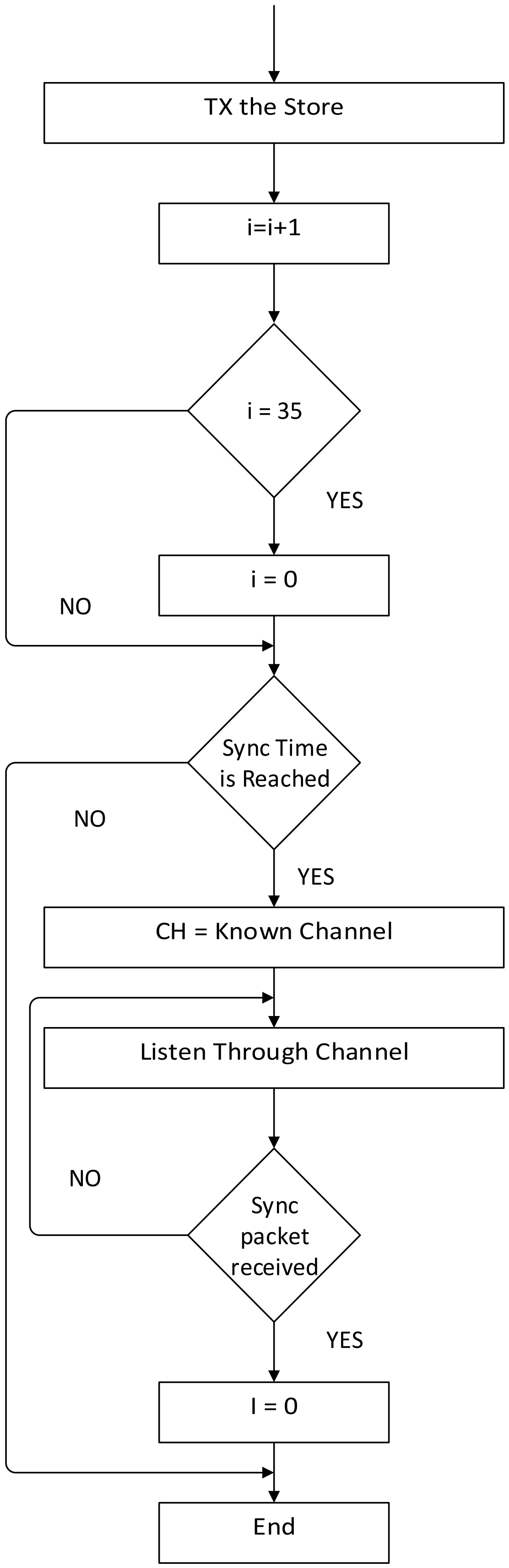
4.3.2 Internals and Working of Transmitter

On boot up, synchronization is required. Nordic nRF24L01+ has built-in synchronization for its normal working but to apply frequency hopping spread spectrum technique, synchronization of frequencies is also required which is proposed in this thesis. According to the mechanism of this synchronization, at startup, a known channel is used by transmitter and receiver. Transmitter works as receiver and searches for sync request packet from receiver which is working as transmitter at boot. As sync packet is received, transmitter resets its counter and switches to its original mode which is to work as transmitter. Now channel is selected from predefined pseudo random channel sequence. In this system this sequence is generated by using Fortuna PRNG and is known to both, transmitter and receiver. Nordic nRF24L01+ converts this number to corresponding frequency. Frequency range varies between 2400-2525 MHz. It has 126 channels and each channel represents a frequency as channel 0 represents 2400 MHz, 1 represents 2401 MHz and so on. In proposed mechanism, channel 0 is used for synchronization. Remaining 125 channels are used for hopping. Channel is changed to next channel after every 10ms to achieve a hop rate of 100 hops/sec. Channel counter “i” points to the location of randomly placed channels and is used for selection of channels. MCU reads sensor array, apply modified AES-256 encryption on captured data and transmits using Nordic nRF24L01+. Even if, we are using same microcontroller at receiver and transmitter, they have minute difference in their clocks which causes the system out of synchronization after some time due to high hop rate. To overcome this problem, after every 10 seconds, synchronization process is repeated. During this, transmitter and receiver follow same procedure as discussed above and reset their counters.

Complete proposed transmitter design is shown in following flow graph.

Transmitter Flow Graph:




TX the Store
i=i+1
i = 35
YES
NO
i = 0
Sync Time is Reached
NO
YES
CH = Known Channel
Listen Through Channel
NO
Sync packet received
YES
I = 0
End

4.3.3 Receiver Design

Like transmitter, this part of system is also consisted of four main parts. Transmitted data is received at receiver side by using Nordic nRF24L01+ which is an excellent transceiver and has good communication rang in Industrial Scientific and Medical (ISM) band which is freely available for use. Additionally, this module works at low power. This transceiver is mounted on microcontroller. Received data is captured in array and then it is sent towards microcontroller for further processing. A pre shared key is used to decrypt the received data at this point. After decryption, received data is showed on LCD attached with microcontroller. Like transmitter, power unit is included with the receiver system also which is a 5V recharge able DC battery.

The block diagram of the proposed transmitter subsystem is shown in Figure 4.2.

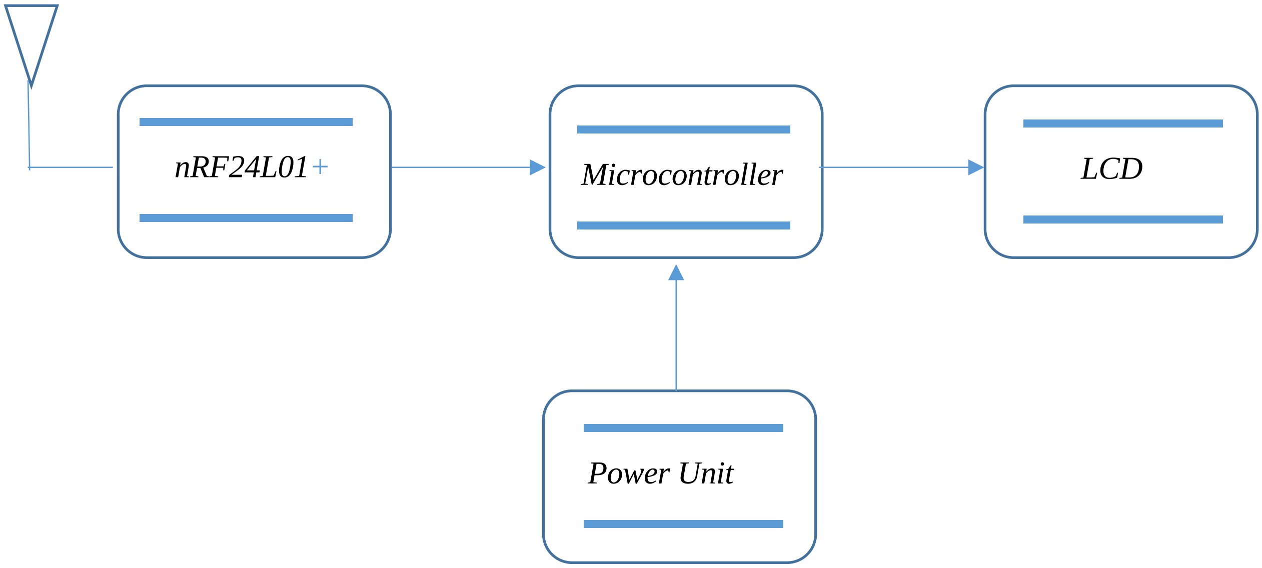


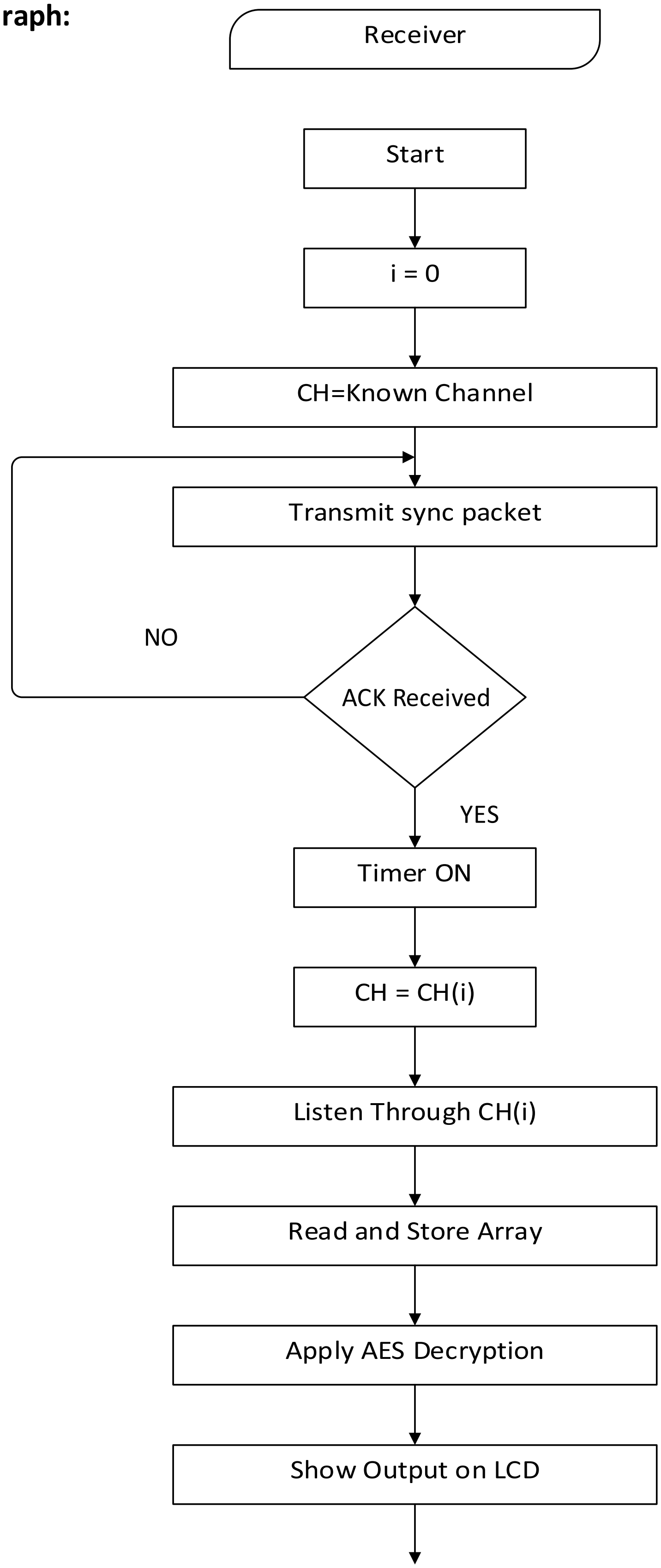
Figure 4.2: Receiver Subsystem

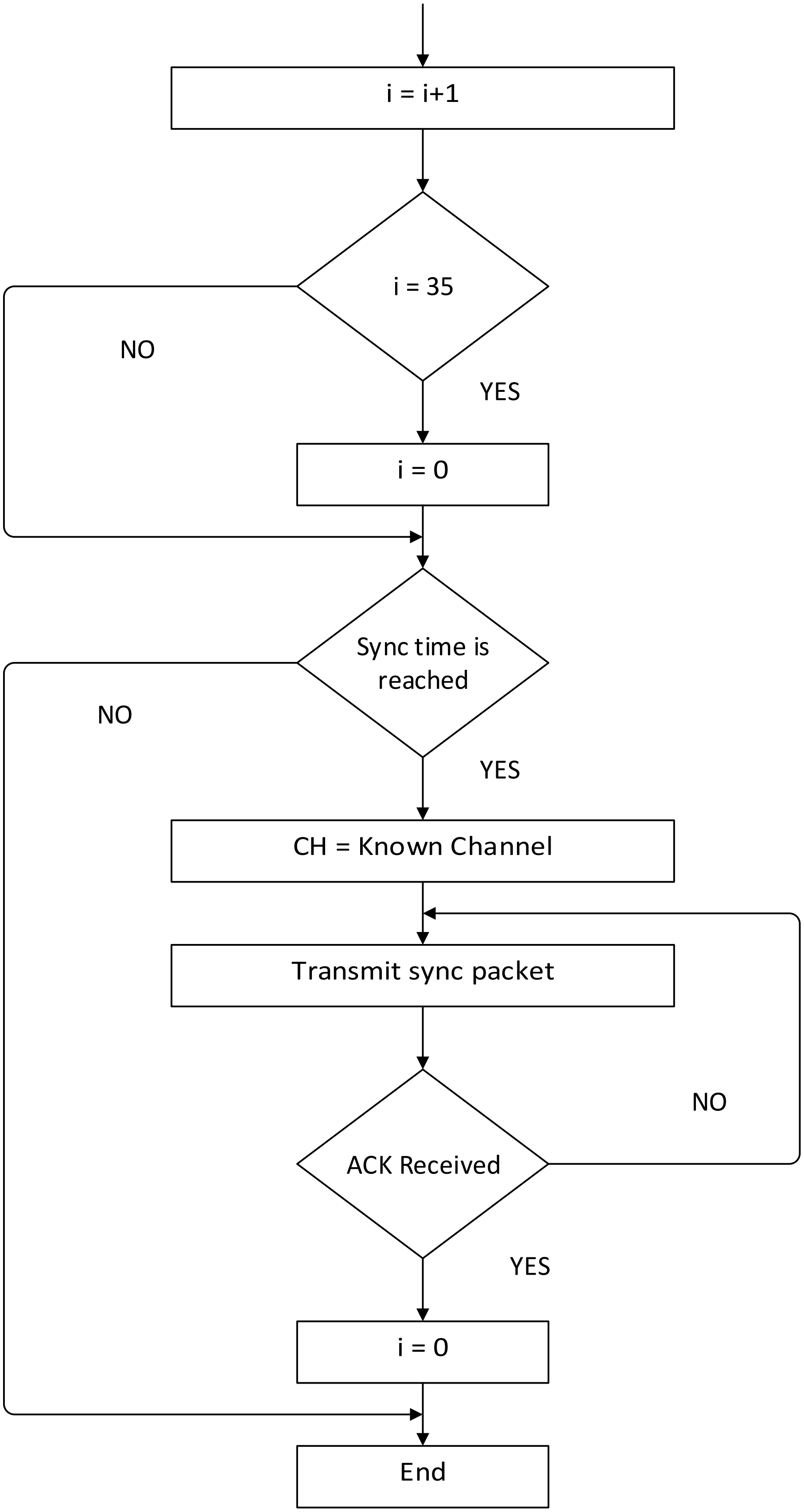
4.3.4 Internals and Working of Receiver

Like transmitter, synchronization is also mandatory for receiver. For this purpose, receiver behaves as transmitter and sends synchronization packet. When this synchronization packet is captured by actual transmitter, which is working as receiver before synchronization, it transmits an acknowledgement towards receiver. On reception of acknowledgement, both transceivers switched back to their original states and reset and start their counters to start hopping. Predefined hopping schema obtained by using Fortuna PRNG is placed in microcontrollers of both transmitter and receiver. This schema is followed by transmitter and receiver to perform hopping. Hopping sequence randomness is key to guarantee link quality of communication. After obtaining synchronization, transmitter starts to transmit encrypted data on hopped channels. This data is captured by Nordic nRF24L01+ and fed to microcontroller for further processing. Microcontroller receives the data in form of array and uses a predefined key of 256 bits to decrypt the data. A liquid crystal library is used to connect a 16x2 LCD to microcontroller. Decrypted output is displayed on LCD attached to microcontroller. After every 10 seconds synchronization will be performed to overcome clock issue as discussed above. Complete procedure of synchronization is followed for resynchronization and it is seem less in proposed system i.e. link of communication does not break during resynchronization of transmitter and receiver.

Internal structure of receiver is described in following flow graph

Receiver Flow Graph:

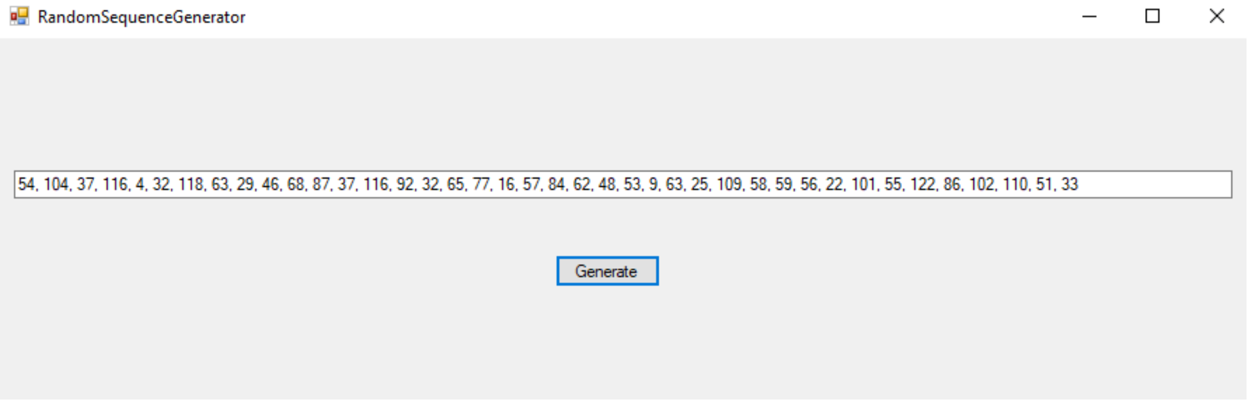



i = i+1
i = 35
NO
YES
i = 0
Sync time is reached
NO
YES
CH = Known Channel
Transmit sync packet
NO
ACK Received
YES
i = 0
End

4.3.5 Random Sequence Generator

Random sequence is required by transmitter and receiver and it should be known by both of them. After synchronization, this pattern is followed by transmitter and receiver and should be highly randomized to avoid interference from other nodes. To generate this random sequence, a well-known algorithm, named Fortuna Pseudo Random Number Generator for Linux operating system is used. To use this algorithm, slight modifications in Fortuna were made to port it for Windows Platform. It is available as open source but it is for Linux operating system only. While porting for Visual Studio 2017, initial entropy for Fortuna is taken from Microsoft Crypto API which in case of Linux was taken from OS data from file "/dev/urandom". This is not available in Windows so crypto API included in .NET Framework is used to serve the matter. Only initial data is captured using this API and rest of Fortuna algorithm is not altered. A user friendly Windows application is made in Visual Studio 2017 using Fortuna PRNG at backend for ease of users.

Figure 4.3 shows GUI of Random Sequence Generator.



```
54, 104, 37, 116, 4, 32, 118, 63, 29, 46, 68, 87, 37, 116, 92, 32, 65, 77, 16, 57, 84, 62, 48, 53, 9, 63, 25, 109, 58, 59, 56, 22, 101, 55, 122, 86, 102, 110, 51, 33
```

Figure 4.3: Random Sequence Generator

Chapter 5

Results and Analysis

5.1 Overview

This chapter demonstrates the results which were obtained during the course of the research. First part of this chapter describes the results of COMSEC part of this research. The results of RF end of proposed system using Spectrum Analyzer and SDR based spectrum analyzer are discussed in second part of this chapter.

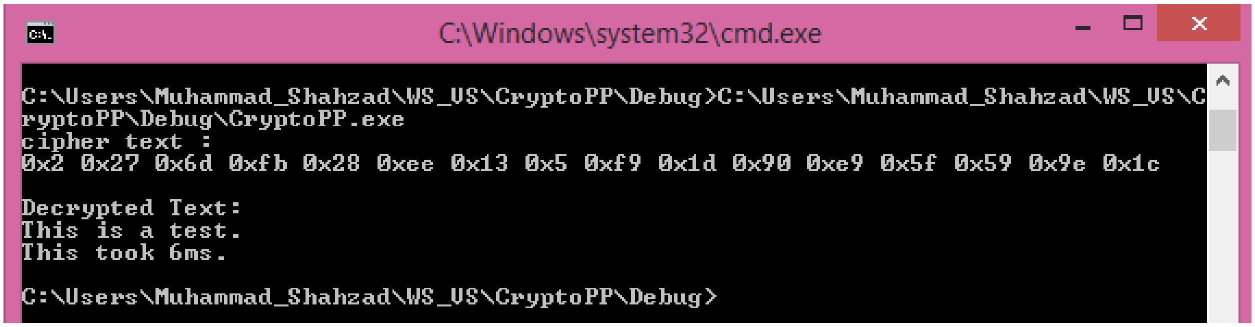
5.2 COMSEC

HP ProBook 4530s laptop with Intel Core i7 2.20 GHz, second generation, 8 GB RAM with Windows 8.1 64 bit is used to simulate COMSEC algorithm. A simulator has been written in C++ to find the robustness of algorithms by making use of high end clock. Microsoft Visual Studio 2017 is used to as developing environment. A comparison of robustness is made between our implementation and well known best open source implementations of cipher i.e. OpenSSL and CryptoPP.

5.2.1 AES-256 Cipher of CryptoPP

While AES-256 Cipher of CryptoPP was simulated on above described platform, to encrypt and decrypt 16 bytes of plaintext, it took 6ms to perform the operation.

Figure 5.1 shows the simulation results of AES-256 Cipher of CryptoPP.



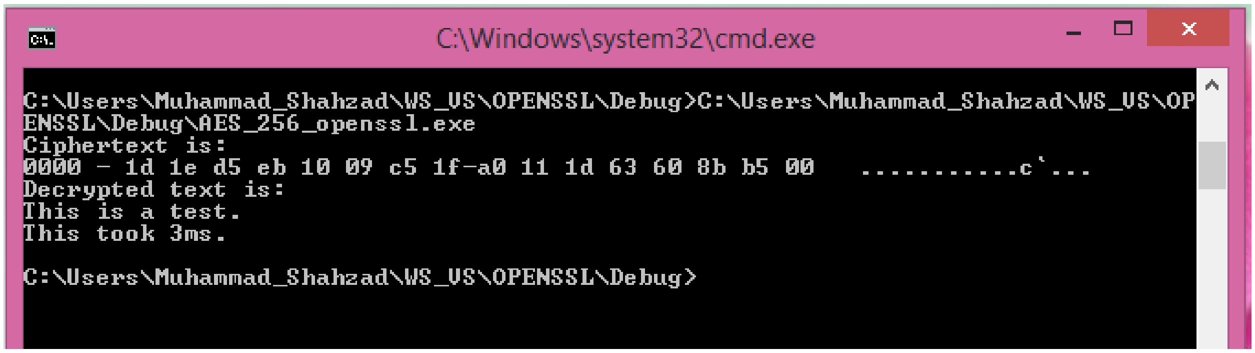
```
C:\Users\Muhammad_Shahzad\WS_US\CryptoPP\Debug>C:\Users\Muhammad_Shahzad\WS_US\C
ryptoPP\Debug\CryptoPP.exe
cipher text :
0x2 0x27 0x6d 0xfb 0x28 0xee 0x13 0x5 0xf9 0x1d 0x90 0xe9 0x5f 0x59 0x9e 0x1c

Decrypted Text:
This is a test.
This took 6ms.

C:\Users\Muhammad_Shahzad\WS_US\CryptoPP\Debug>
```

Figure 5.1: Robustness of AES-256 Cipher of CryptoPP

5.2.2 AES-256 Cipher of OpenSSL

OpenSSL is another well-known, open source and efficient implementation of ciphers. While AES-256 Cipher of latest version of OpenSSL by July 2018 was simulated on above described platform, and 16 bytes of plaintext were encrypted and decrypted, they took 3ms to completely perform the operation.

Figure 5.2 shows the simulation results of AES-256 Cipher of OpenSSL.



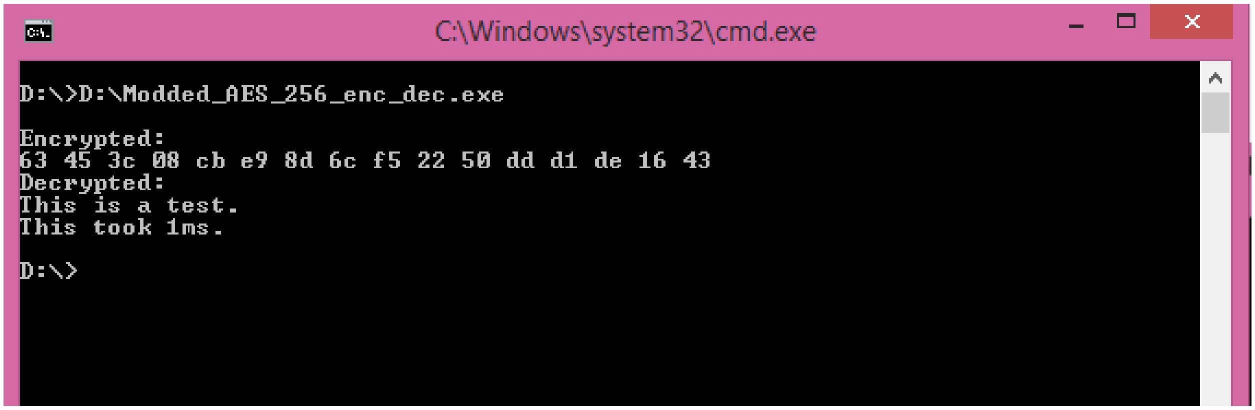
```
C:\Users\Muhammad_Shahzad\WS_US\OPENSSL\Debug>C:\Users\Muhammad_Shahzad\WS_US\OP
ENSSL\Debug\AES_256_openssl.exe
Ciphertext is:
0000 - 1d 1e d5 eb 10 09 c5 1f-a0 11 1d 63 60 8b b5 00   ...........c`...
Decrypted text is:
This is a test.
This took 3ms.

C:\Users\Muhammad_Shahzad\WS_US\OPENSSL\Debug>
```

Figure 5.2: Robustness of AES-256 Cipher of OpenSSL

5.2.3 Proposed Implementation of AES-256 Cipher

While proposed Implementation of AES-256 cipher was simulated on above described platform, and 16 bytes of plaintext were encrypted and decrypted, they took 1ms to completely perform the operation.

Figure 5.3 shows the simulation results of proposed Implementation of AES-256 Cipher.



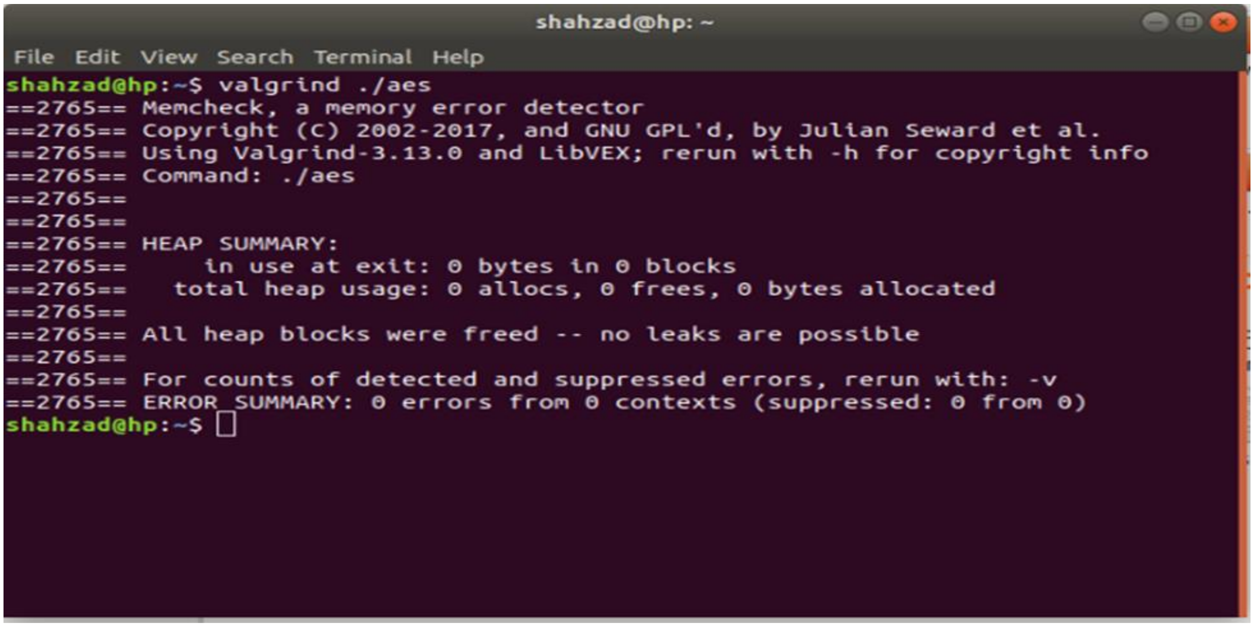
```
D:\>D:\Modded_AES_256_enc_dec.exe

Encrypted:
63 45 3c 08 cb e9 8d 6c f5 22 50 dd d1 de 16 43
Decrypted:
This is a test.
This took 1ms.

D:\>
```

Figure 5.3: Robustness of Proposed Implementation of AES-256 Cipher

5.2.4 Valgrind Results of Proposed Implementation of AES-256 Cipher

Memory leak test of proposed Implementation of AES-256 cipher is performed on open source Linux based tool named Valgrind. Test result shows that no memory leaks are present in proposed Implementation of AES-256 cipher.

Figure 5.4 shows the results of memory test.



```
shahzad@hp:~$ valgrind ./aes
==2765== Memcheck, a memory error detector
==2765== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==2765== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==2765== Command: ./aes
==2765==
==2765==
==2765== HEAP SUMMARY:
==2765==     in use at exit: 0 bytes in 0 blocks
==2765==   total heap usage: 0 allocs, 0 frees, 0 bytes allocated
==2765==
==2765== All heap blocks were freed -- no leaks are possible
==2765==
==2765== For counts of detected and suppressed errors, rerun with: -v
==2765== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
shahzad@hp:~$
```

Figure 5.4: Memory Test Results

5.3 TRANSEC

This section shows the spectrum analysis of proposed system. Functionality of frequency hopping algorithm is verified using more precise and accurate spectrum analyzers. Software Defined Radio techniques are used to examine frequency spectrum of proposed mechanism. Universal Serial Radio Peripheral (USRP 2) is used along with GNU Radio and gr-fosphor to precisely analyze the channel bandwidth of proposed system.

5.3.1 Spectrum of Proposed System

Spectrum of proposed system was analyzed to verify the working of purposed algorithm. ISM band is used for transmission. A hop rate of 100 hops/sec. is achieved successfully by using proposed transmission security algorithm and accurate synchronization mechanism. Frequencies between 2400-2525 MHz are utilized to carry out the transmission of data and 2400 MHz is reserved for synchronization according to purposed mechanism of synchronization. The Digital RF-Explorer 3G handheld combo spectrum analyzer is used to analyze the spectrum of proposed system. RF Explorer trusts on frequency synthesizer which is highly integrated and has a mixer with double balance.

Spectrum of proposed system, captured by using Touchstone Pro Windows software, is displayed in figure 5.5.

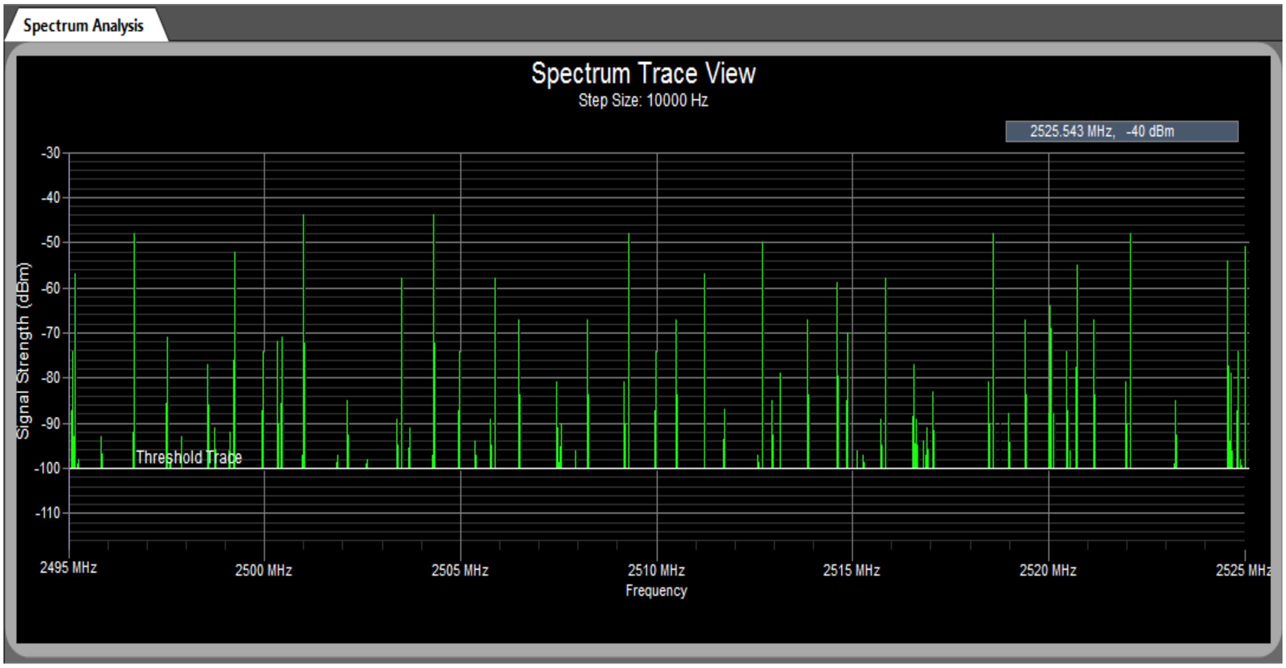


Figure 5.5: Frequency Spectrum of Proposed System

5.3.2 Channel Bandwidth and Waterfall Using gr-fosphor

GR-fosphor is used to observe channel bandwidth of proposed system. Careful reading of frequency spectrum shows that 1 MHz of bandwidth is used by the proposed system. Figure 5.6 shows the bandwidth utilized by the proposed implementation. Waterfall plot of the implemented system is also shown in figure 5.6. It is obvius that as data is transmitted, a horizontal line appears on real time waterfall plot.

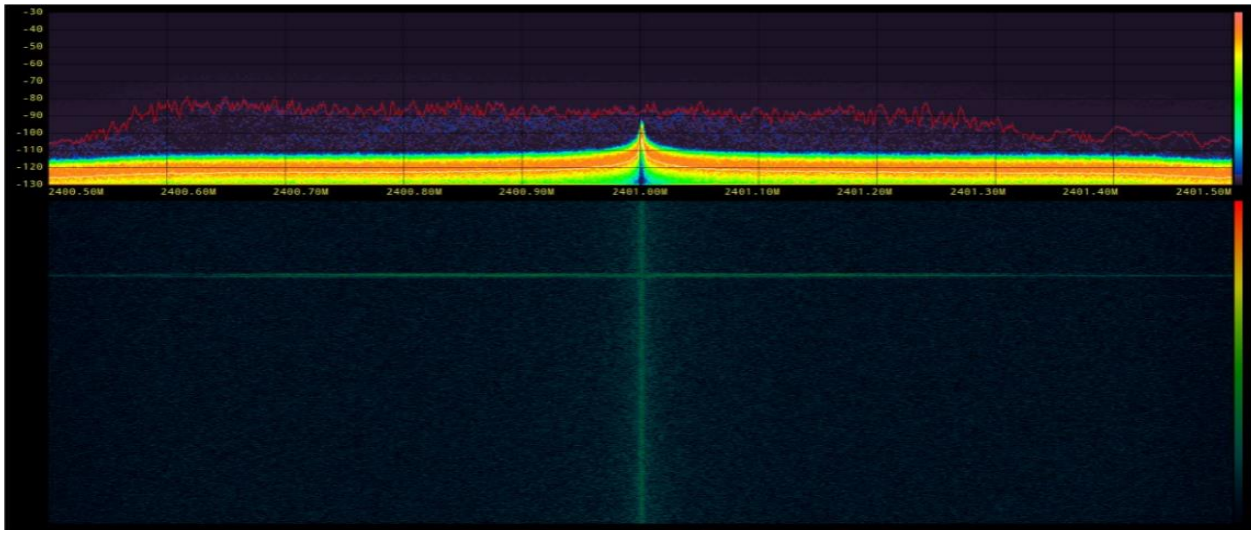
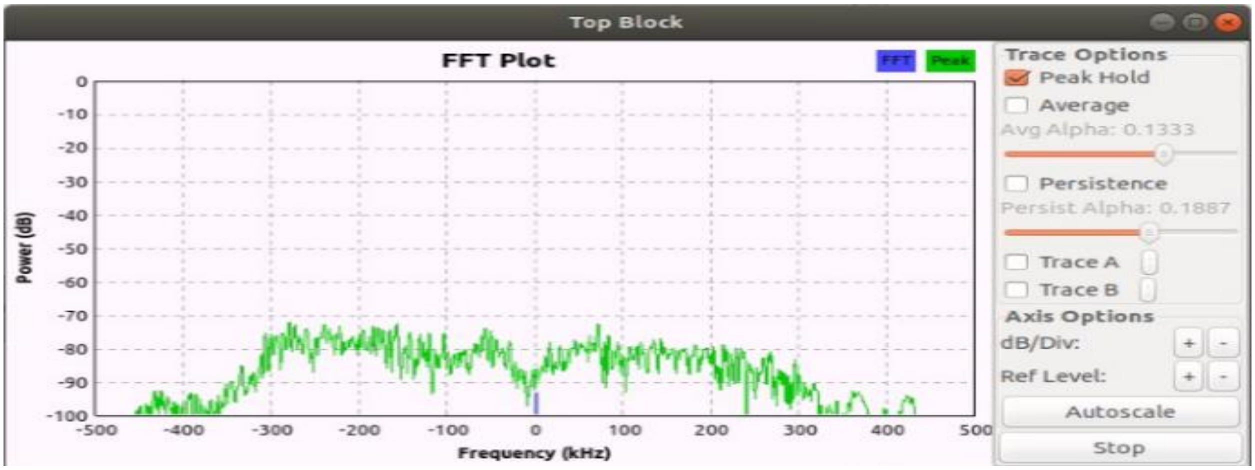


Figure 5.6: Channel Bandwidth and Waterfall Plots Using gr-fosphor

5.3.3 Channel Bandwidth and Waterfall using WX GUI FFT Sink

Figure 5.7 shows the results obtained by using USRP 2 and built-in GNU Radio Block FFT Sink.

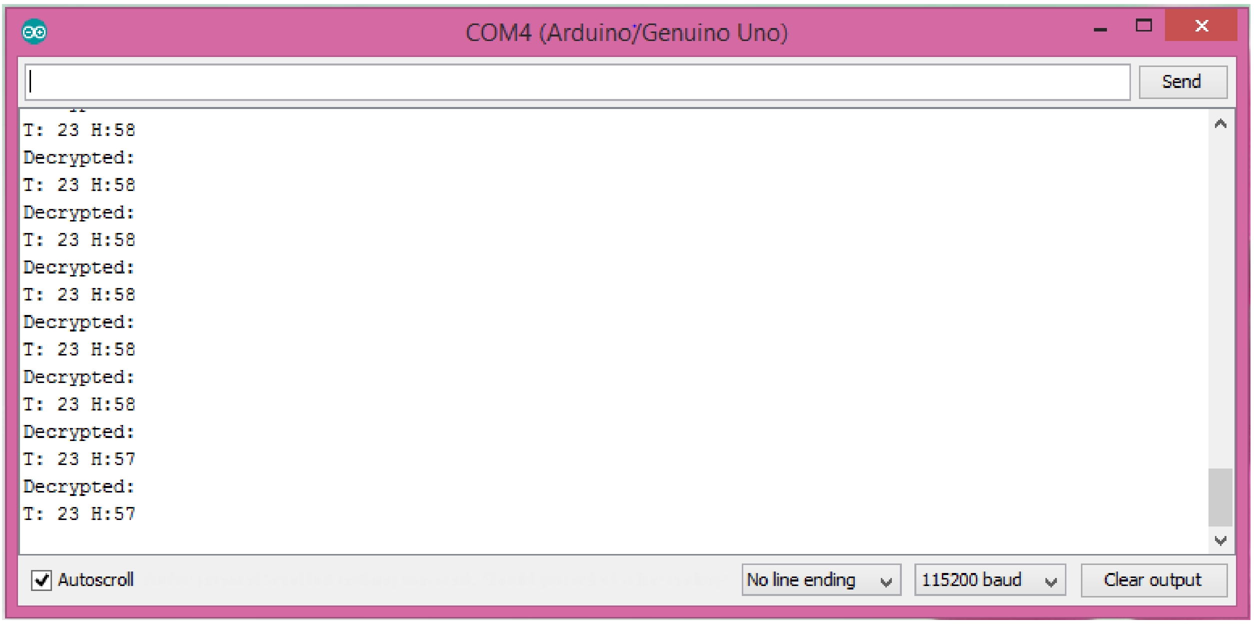


☑ Peak Hold
□ Average
Avg Alpha: 0.1333
□ Persistence
Persist Alpha: 0.1887
□ Trace A
□ Trace B

Figure 5.7: Channel Bandwidth Using FFT Sink

5.3.4 Serial Monitor Results

When receiver side is connected to Windows PC through serial monitor of Arduino IDE, the received decrypted data is shown on serial monitor as displayed in Figure 5.8.



```
T: 23 H:58
Decrypted:
T: 23 H:58
Decrypted:
T: 23 H:58
Decrypted:
T: 23 H:58
Decrypted:
T: 23 H:58
Decrypted:
T: 23 H:58
Decrypted:
T: 23 H:57
Decrypted:
T: 23 H:57
```

☑ Autoscroll   No line ending   115200 baud

Figure 5.8: Serial Monitor Output

5.3.5 16x2 LCD Results

When receiver side is connected to a 16x2 LCD, the received decrypted data is shown on LCD as displayed in Figure 5.9.

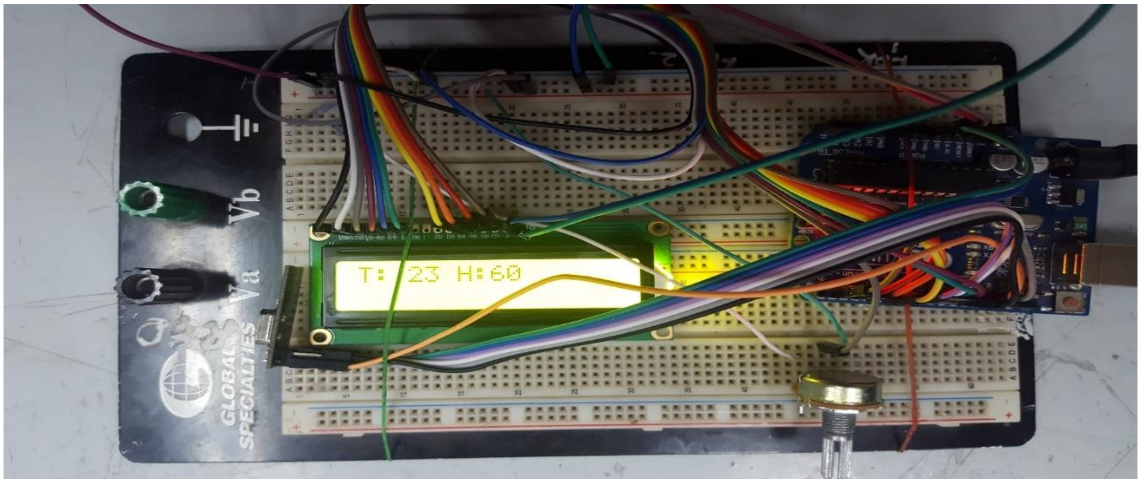


Figure 5.9: 16x2 LCD Output

5.4 Analysis

Proposed system is reluctant against security attacks at application layer and physical layer. A comprehensive analysis of COMSEC and TRANSEC for WSNs has been carried out to implement the most suitable for desired application. This analysis describes the attacks launched on all layers of WSNS and their proposed solutions. Most of presented solutions of application layer and physical layer attacks, require intensive processing and some of them are weaker in terms of security. In this research, a strong cipher known as AES-256 is selected to perform optimizations to make it robust and light weight so that it could be used on low cost and limited resource microcontrollers. A comparison of robustness of proposed implementation and currently present most efficient implementation i.e. OpenSSL and CryptoPP, is made. This comparison shows that proposed algorithm is three times robust than OpenSSL and six times robust than CryptoPP. Memory leak is very important when we are working on low resource hardware components i.e. microcontrollers. Proposed implementation is tested using Valgrind in Linux OS to check memory leaks present in it. These results shows that no leaks of memory are possible in this implementation.

For TRANSEC, a light weight frequency hopping algorithm and its synchronization mechanism is proposed and implemented using ATmega328 and multichannel RF transceiver of WSNs. Algorithm is successfully implemented and spectrum is observed using RF explorer handheld spectrum analyzer and Touchstone Pro Windows software. Spectrum shows that frequency is hopping between 2401-2525 MHz and 2400 MHz is reserved for synchronization. Waterfall plot is observed using gr-fosphor in GNU Radio. USRP 2 is used in GNU Radio as RF end to capture the spectrum. Channel bandwidth is observed to be 1 MHz, which is verified by using WX GUI Sink of GNU Radio. Data is transmitted and received successfully and hop rate of 100 hops/sec. is observed.

Epitome

6.1 Research Achievements

A comprehensive analysis of COMSEC and TRANSEC has been carried out for WSNs to implement the most suitable for desired application. A secure and reliable communication framework for WSNs is developed. AES 256 bit for Communication Security has been implemented on microcontroller. Modifications in COMSEC have been done to enhance the speed of processing and reduce the amount of resources on microcontroller. For TRANSEC, a mechanism of frequency hopping and its synchronization has been proposed and implemented to achieve hop rate of 100 hops/sec for WSNs.

6.2 Application Area

Better confidentiality with less resource utilization, high reliability, low cost, enhanced anti-jamming performance and less power consumption make this research suitable to be used in various applications of WSNs in ISM band. This modified fast implementation of Advanced Encryption Standard (AES-256) can be used on any low cost limited resources microcontrollers. Designed algorithm of frequency hopping and its synchronization is best choice to be used in WSNs which have limited amount of computational resources.

6.3 Future Work

Though this thesis presents powerful COMSEC implementation with enough robustness that it can work on limited resource microcontroller. This approach can be applied to other encryption algorithms i.e. Camellia-256 to make it robust so that it can be used for WSNs applications. Secondly, time correction offset based synchronization can be implemented for proposed algorithm to improve its performance even more.

Installing gr-fosphor on Linux

1) Install dependencies

```
sudo apt-get install cmake xorg-dev libglu1-mesa-dev
```

2) Build GLFW

```
git clone https://github.com/glfw/glfw

cd glfw

mkdir build

cd build

cmake ../ -DBUILD_SHARED_LIBS=true

make

sudo make install

sudo ldconfig
```

3) Install OpenCL

```
sudo apt-get install nvidia-opencl-dev opencl-headers
```

4) Install qt-sdt

```
sudo apt install qt-sdk
```

5) Install osmosdr

```
sudo apt install gr-osmosdr
```

6) Install gr-fosphor

```
git clone git://git.osmocom.org/gr-fosphor

cd gr-fosphor

mkdir build

cd build

cmake ../

make

sudo make install

sudo ldconfig
```

7) Install GNU Radio

```
sudo apt-get install gnuradio
```

8) Install UHD images

```
sudo uhd_images_downloader
```

9) Manually configure IP address of USRP 2 to

192.168.10.1

Subnet mask

255.255.255.0

Installing Valgrind on Linux

1) Go to following link to download latest release of Valgrind and place it in home directory

http://valgrind.org/downloads/current.html

2) Install dependencies by using following command in terminal

```
sudo apt-get install libc6-dbg gdb
```

3) Use following commands to install valgrind

```
tar jxvf valgrind-3.13.0.tar.bz2

cd valgrind-3.13.0.tar.bz2

./configure

make

sudo make install
```

Building and testing the code

```
gcc aes.c -o aes

valgrind ./aes
```

Appendix-III

Source Code of Simulator

```
#include <chrono>
#include <ctime>
#include<iostream>

template <typename T> class basic_stopwatch
{
       typedef T clock;
       typename clock::time_point p;
       typename clock::duration   d;

public:
       void tick() { p = clock::now(); }
       void tock() { d += clock::now() - p; }
       void reset() { d = clock::duration::zero(); }

       template <typename S> unsigned long long int report() const
       {
              return std::chrono::duration_cast<S>(d).count();
       }

       unsigned long long int report_ms() const
       {
              return report<std::chrono::milliseconds>();
       }

       basic_stopwatch() : p(), d() { }
};

struct c_clock
{
       typedef std::clock_t time_point;
       typedef std::clock_t duration;
       static time_point now() { return std::clock(); }
};

template <> unsigned long long int basic_stopwatch<c_clock>::report_ms() const
{
       return 1000. * double(d) / double(CLOCKS_PER_SEC);
}

typedef basic_stopwatch<std::chrono::high_resolution_clock> stopwatch;
typedef basic_stopwatch<c_clock> cstopwatch;
```

Usage

```
       stopwatch sw;
       sw.tick();

       //Processing Here


       sw.tock();
       cout << endl << "This took " << sw.report_ms() << "ms.\n";
```

REFERENCES

[1] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” Comput. Networks, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] W. Dargie and C. Poellabauer, “FUNDAMENTALS OF WIRELESS SENSOR NETWORKS THEORY AND PRACTICE.”

[3] J. M. Corchado, J. Bajo, D. I. Tapia, and A. Abraham, “Using Heterogeneous Wireless Sensor Networks in a Telemonitoring System for Healthcare,” IEEE Trans. Inf. Technol. Biomed., vol. 14, no. 2, pp. 234– 240, Mar. 2010.

[4] J. Gutierrez, J. F. Villa-Medina, A. Nieto-Garibay, and M. A. Porta- Gandara, “Automated Irrigation System Using a Wireless Sensor Network and GPRS Module,” IEEE Trans. Instrum. Meas., vol. 63, no. 1, pp. 166– 176, Jan. 2014.

[5] K. K. Khedo, R. Perseedoss, A. Mungur, U. of Mauritius, and Mauritius, “A Wireless Sensor Network Air Pollution Monitoring System,” May 2010.

[6] Y. E. Aslan, I. Korpeoglu, and Ö. Ulusoy, “A framework for use of wireless sensor networks in forest fire detection and monitoring,” Comput. Environ. Urban Syst., vol. 36, no. 6, pp. 614–625, 2012.

[7] Al Ameen, M.; Liu, J.; Kwak, K. Security and privacy issues in wireless sensor networks for healthcare applications. J. Med. Syst. 2012, 36, 93–101.

[8] Undercoffer, J.; Avancha, S.; Joshi, A.; Pinkston, J. Security for sensor networks. In Proceedings of the CADIP Research Symposium, Baltimore, MD, USA, 29–31 October 2002.

[9] W. Diffie and M. E. Hellman, “New Directions in Cryptography”, IEEE Transaction on Informa- tion Theory, IT-22:644C654, November 1976.

[10] R. L. Rivest, A. Shamir, and L. M. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," Communications of the ACM, 21(2):120-126, 1978.

[13] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," Comput. Networks, vol. 52, no. 12, pp. 2292–2330, 2008.

[14] M. A. Jan, "Energy-efficient Routing and Secure Communication in Wireless Sensor Networks," no. February, p. 209, 2016.

[15] J. Kumari and Prachi, "A comprehensive survey of routing protocols in wireless sensor networks," pp. 325–330, 2015.

[16] S. P. Dongare and R. S. Mangrulkar, "Implementing Energy Efficient Technique for Defense against Gray-Hole and Black-Hole Attacks in Wireless Sensor Networks," pp. 167–173, 2015.

[17] S. Kaplantzis, "Security Models for Wireless Sensor Networks," PhD Convers. Report, Cent. Telecommun. Inf. Eng. Monash Univ. Aust., 2006.

[18] Shi, E.; Perrig, A. Designing secure sensor networks. IEEE Wirel. Commun. 2004, 11, 38–43.

[19] Wang, Y.; Attebury, G.; Ramamurthy, B. A survey of security issues in wireless sensor networks. IEEE Commun. Surv. Tutor. 2006, 8, 1–21. [CrossRef]

[20] Xu, W.; Ma, K.; Trappe, W.; Zhang, Y. Jamming sensor networks: Attack and defense strategies. IEEE Netw. 2006, 20, 41–47.

[21] Pelechrinis, K.; Iliofotou, M.; Krishnamurthy, S.V. Denial of service attacks in wireless networks: The case of jammers. IEEE Commun. Surv. Tutor. 2011, 13, 245–257. [CrossRef]

[22] Anderson, R.; Kuhn, M. Tamper resistance: A cautionary note. In Proceedings of the Second USENIX Workshop on Electronic Commerce, Oakland, CA, USA, 18–20 November 1996.

[23] Jokhio, S.H.; Jokhio, I.A.; Kemp, A.H. Node capture attack detection and defence in wireless sensor networks. IET Wirel. Sens. Syst. 2012, 2, 161–169. [CrossRef]

[24] Giannetsos, T.; Dimitriou, T. Spy-Sense: Spyware tool for executing stealthy exploits against sensor networks. In Proceedings of the 2nd ACM Workshop on Hot Topics on Wireless Network Security and Privacy, Budapest, Hungary, 17–19 April 2013; pp. 7–12.

[25] Barenghi, A.; Breveglieri, L.; Koren, I.; Naccache, D. Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures. IEEE Proc. 2012, 100, 3056–3076. [CrossRef]

[26] Karri, R.; Wu, K.; Mishra, P.; Kim, Y. Fault-based side-channel cryptanalysis tolerant Rijndael symmetric block cipher architecture. In Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, San Francisco, CA, USA, 24–26 October 2001; pp. 427–435.

[27] Amiel, F.; Villegas, K.; Feix, B.; Marcel, L. Passive and active combined attacks: Combining fault attacks and side channel analysis. In Proceedings of the Workshop on Fault Diagnosis and Tolerance in Cryptography, Vienna, Austria, 10 September 2007.

[28] Wood, A.D.; Stankovic, J.A. Denial of service in sensor networks. Computer 2002, 35, 54–62. [CrossRef]

[29] Noda, C.; Prabh, S.; Alves, M.; Voigt, T. On packet size and error correction optimisations in low-power wireless networks. In Proceedings of the 10th Annual IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks, New Orleans, LA, USA, 24–27 June 2013.

[30] Karlof, C.; Wagner, D. Secure routing in wireless sensor networks: Attacks and countermeasures. Ad Hoc Netw. 2003, 1, 293–315. [CrossRef]

[31] Newsome, J.; Shi, E.; Song, D.; Perrig, A. The Sybil attack in sensor networks: Analysis & defenses. In Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, Berkeley, CA, USA, 26–27 April 2004.

[32] Singh, V.P.; Sweta, J.; Jyoti, S. Hello flood attack and its countermeasures in wireless sensor networks. Int. J. Comput. Sci. 2010, 7, 23–27.

[33] Zhao, H.; Li, Y.; Shen, J.; Zhang, M.; Zheng, R.; Wu, Q. A new secure geographical routing protocol based on location pairwise keys in wireless sensor networks. Int. J. Comput. Sci. Issues 2013, 10, 365-372.

[34] Juels, A.; Brainard, J.G. Client puzzles: A cryptographic countermeasure against connection depletion attacks. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 3 February 1999.

[35] Raymond, D.R.; Midkiff, S.F. Denial-of-service in wireless sensor networks: Attacks and defenses. IEEE Pervasive Comput. 2008, 7, 74–81. [CrossRef]

[36] Deng, J.; Han, R.; Mishra, S. Defending against path-based DoS attacks in wireless sensor networks. In Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, Alexandria, VA, USA, 7–10 November 2005.

[37] Karlof, C.; Sastry, N.; Wagner, D. TinySec: A link layer security architecture for wireless sensor networks. In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, Baltimore, MD, USA, 3–5 November 2004; pp. 162–175.

[38] Luk, M.; Mezzour, G.; Perrig, A.; Gligor, V. MiniSec: A secure sensor network communication architecture. In Proceedings of the 6th International Symposium on Information Processing in Sensor Networks, Cambridge, MA, USA, 22–24 April 2007; pp. 479–488.

[39] Shiva Murthy G, Robert John D'Souza and Golla Varaprasad "Digital Signature-Based Secure Node disjoint Multipath Routing Protocol for Wireless Sensor Networks", IEEE Sensors Journal, VOL. 12, Issue-10, 2012, pp.2941- 2949.

[40] G.Rohini, “Dynamic Router Selection and Encryption for Data Secure in Wireless Sensor Networks, Information Communication and Embedded Systems (ICICES), IEEE 2013 International Conference on , 2006, pp. 256 - 259.

[41] Sangeetha R. and Yuvaraju M. “Secure Energy-Aware Multipath Routing Protocol with Transmission Range Adjustment for Wireless Sensor Networks”, Computational Intelligence & Computing Research (ICCIC), 2012 IEEE International Conference on, 2012, pp. 1-4.

[42] Suraj Kumar Sharma and Sanjay Kumar Jena, “SCMRP: Secure Cluster Based Multipath Routing Protocol for Wireless Sensor Networks,” Sixth International Conference on Wireless Communication and Sensor Networks (WCSN), IEEE 2010, pp. 1-6.

[43] Guiyi Wei and Xueli Wang “Detecting Wormhole Attacks Using Probabilistic Routing and Redundancy Transmission,” International Conference on Multimedia Information Networking and Security, IEEE 2010, pp. 496- 500.

[44] C.Paar and J.Pelzl, Understanding Cryptography: a textbook for students and practitioners. Berlin: Springer, 2010.

[45] L. Peterson, R. E. Ziemer, D. E. Borth, “Introduction to Spread Spectrum Communications”, Prentice-Hall, N.J, April 1995. ISBN-10: 0024316237. ISBN-13: 978-0024316233.

[46] K.H.Torvmark,”Frequency Hopping Systems”, Application Note AN014. Chipcon Products from Texas Instruments Inc, March 2002.

[47] B. Sklar, “Digital Communications, Fundamentals and Applications”, 2nd Ed. Prentice-Hall Inc., Upper Saddle River, N.J, 2001. ISBN: 0-13-084788-7.

[48] T. Vanninen, H. Saarmisaari, M.Rustia, and T. Koskela, “FH-Code Phase Synchronization in a Wireless Multi-Hop ADHOC Network”, milcom, MILCOM 2006, pages1-7.

[49] D. Torrieri, “Principles of Spread Spectrum Communications systems”. Springer Science + Business Media Inc., 2005. ISBN: 0-387-22782-2.

[50] V. P. Ipatov, “Spread Spectrum and CDMA: Principles and Applications”, John Wiley & Sons Ltd, Chichester, England, 2005. ISBN: 0-470-09178-9 (HB).

[51] F. Dominique and J.H. Reed, “Robust Frequency Hop Synchronization Algorithm”, Electronics Letters, Vol. 32, No.16, August 1996, pages 1450-1451.

[52] Wang Y, Guardiola I G, Wu X. RSSI and LQI Data Clustering Techniques to Determine the Number of Nodes in Wireless Sensor Networks. International Journal of Distributed Sensor Networks, 2014, 2014(6, article 135):469-470.

[53] Wang J Q, Wu J. An Improved Frequency Agility Mechanism for SimpliciTI Protocol. Applied Mechanics & Materials, 2014, 543-547:3486-3489.

[54] Shi Xin, Yin Aimin, Chen Xi. RSSI and multidimensional scaling based indoor localization algorithm. Chinese Journal of Scientific Instrument, 2014, 35(2):261-268.

[55] Wu Shilong, Zhang Wanli Yang Xiaoying,.RSSI modified WSN localization algorithm. Journal of Chongqing University, 2014, 37 (08): 144-150.

[56] Schaefer F, Kays R. Frequency Hopping for Indoor Fading Channels With Varying Level of Environmental Mobility. IEEE Wireless Communication Letters, 2015, 4(1):42-45.

[57] Zoppi S, Gürsu H M, Vilgelm M, et al. Reliable Hopping Sequence Design for Highly Interfered Wireless Sensor Networks[C]// The, IEEE International Symposium on Local and Metropolitan Area Networks. IEEE, 2017.

[58] Chen Zhenping, Li Dequan, Huang Yourui, Tang Chaoli, Li Peng. Mixed-triggered consensus time synchronization

[59] Dai Jia, Guo Lili. A fast and high precision frequency hopping synchronous tracking method. Automation technology and application, 2017, 36 (02): 33-36.

[60] Long Haiyan, Zhang Tianfei, Guo Hui, Liang Meiyu, Ding Jiao. Research on the CC2530 Based RSSI Ranging Technology. Electronics World, 2016, (06):168-170.

[61] J.G. Proakis, “Digital Communications”, 4th Ed. The McGraw-Hill companies Inc., N.York, 2000. ISBN: 0-07-232111-3.

[62] Ding Wang. Simulation and analysis of frequency hopping communication system of synchronization technology. Hangzhou Electronic Science and Technology University, 2014.

[63] C. Anton-Haro and M. Dohler, “Introduction to machine-to- machine (M2M) communications,” in Machine-to-machine (M2M) Communications Architecture, Performance and Applications, C. Anton-Haro and M. Dohler, Eds. Cambridge: Woodhead Publishing, 2015, pp. 27–46.

[64] X. Vilajosana, P. Tuset-Peiro, F. Vazquez-Gallego, J. Alonso- Zarate, and L. Alonso, “Standardized Low-Power Wireless Communication Technologies for Distributed Sensing Applications,” Sensors, vol. 14, no. 2, pp. 2663–2682, 2014.

[65] W. Webb, “Weightless machine-to-machine (M2M) wireless technology using TV white space,” in Machine-to-machine (M2M) Communications Architecture, Performance and Applications, C. Anton-Haro and M. Dohler, Eds. Cambridge: Elsevier, 2015, pp. 93–108.

[66] W. Webb, “Weightless: The Technology to Finally Realise the M2M Vision,” Int. J. Interdiscip. Telecommun. Netw., vol. 4, no. 2, pp. 30–37, 2012.

[67] S. Rao, “Implementing a Bidirectional Frequency Hopping Application with TRF6903 and MSP430”. Application Report. SWRA041, September 2004.

[68] W. Golomb and G. Gong, “Signal Design for Good Correlation: For Wireless Communication, Cryptography, and Radar”, Cambridge University Press, 2005. ISBN: 9780521821049.

[69] http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01. Accessed: 2016-3-10

[70] Singh, G.P. (2014): Designing of a microcontroller based multi-sensor system. Master thesis. Department of Electronics & Communication Engineering, National Institute of Technology, Rourkela, Orissa, India (http://ethesis.nitrkl.ac.in/6041/)

[71] Zhang, P. (2014): Wireless sensor system for monitoring and control. Master thesis. University of VAASA, Faculty of Technology Telecommunication Engineering, Finland (https://www.tritonia.fi/download/gradu/5738)