Republic of Iraq
Ministry of Higher Education
and Scientific Research
University of Anbar
College of Computer
Department of Computer Science

Improving Distributed Public Key Algorithms for E-Health System

A Thesis

Submitted to the Department of Computer Science - College of Computer - University of Anbar as Partial Fulfillment of the Requirement for Master Degree of Science in Computer Sciences

By

Abeer Dawood Salman Alnuaimi

Supervised By

Prof. Dr. Sufyan T. Faraj Al-Janabbi

And

Dr. Ali Jbaeer Dawood

1435 A.H. *2014 A.D.*

سورة القصص

Supervisor Certificate

We certify that this thesis entitled "**Improving Distributed Public key algorithms for E-Health System**" was prepared under my supervision at the Department of computer Science, College of Computer, University of Anbar, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Signature: 

Name: Prof. Dr. Sufyan T. Faraj Al-Janabbi

(Supervisor 1)

Date: / /

Signature: 

Name: Dr. Ali Jbaeer Dawood

(Supervisor 2)

Date: / /

Certificate

I certify that I have read this thesis entitled **"Improving Distributed Public key algorithms for E-Health System"** and I found it linguistically adequate.

Signature:

Name: Assist.Prof. Dr. Ahmed Hameed Ubeid

(Linguist Authority)

Date: ١٤ / ٥ / ٢٠١٤

Examination Committee Certification

We certify that we have read this thesis "**Improving Distributed Public key Algorithms for E-Health System**", and as an Examining Committee the student "**Abeer Dawood Salman**" in its contents and that in our opinion it is adequate to fulfill the requirements for the degree of **Master of Computer Science**.

Signature:

Name: Assist. Prof. Dr. Murtadha Mohammed Hamad (Chairman)

Date: 20/5/2014

Signature: 14/5/2014

Name: Assist. Prof. Dr. Rabah Nory Farhan (Member)

Date: / /

Signature:

Name: Assist. Prof. Dr. Salim Ali Abbas (Member)

Date: 14/5/2014

Signature:

Name: Prof. Dr. Sufyan T. Faraj Al-Janabbi (Supervisor)

Date: / /

Signature:

Name: Dr. Ali Jbaeer Dawood (Supervisor)

Date: / /

The Thesis approved by the College of Computer- University of Anbar.

Signature:

Name: Assist. Prof. Dr. Murtadha Mohamed Hamad (Dean of the College)

Date: 20/5/2014

DEDICATED TO

My Parents…

My Brother…

My Sisters…

My Friends…

To everyone

Taught me

Acknowledgements

All praises to Allah Almighty, the most benevolent and merciful, the creator of universe, who enabled me to complete this work successfully. Thanks to our teacher Muhammad (May Allah Exalt his Mention) who guided us to the right path. I want to express my deepest gratitude to my respectable teacher Dr. Syfyan T. Faraj for his invaluable guidance and support. I would like to thank my respectable teacher Dr. Ali J. Dawood for his guidance in writing this thesis.

Special thanks to Ruqayah R. Al-Dahan for her support and encouragement since the beginning of master studying.

I wish to offer my gratitude to all persons who never hesitated in offering their help when I needed it and for encouraging me from the teaching staff of the college of computer in university of Anbar.

My deepest sense of Acknowledgement goes to my loving parents, and I wish to be Prouder to them. Thanks to my friends Ekram Habeeb and Ketam Abed-Al based who always were supporting me. At the end, I am especially grateful to Ahmed Sulaiman for his Advices.

Abeer Dawood Salman

2013

Abstract

Wireless Body Area Network (WBAN) is an emerging technology used in health care scope, it consists of a number of intelligent sensors that collect the medical data from human body and transfer it by WLAN to the personal device that in turn transfer the patient data to the medical server via the internet where a professional analyzes this data. The back-end server contains sensitive and critical data, and thus it must be enjoyed with high security and privacy. If the patient data could be stolen, tampered, or accessed by any unauthorized person, the data can be lost and even worst altered.

In this thesis, a Distributed Storage System (DSS) is used to provide security and privacy to the patient information through distributing the storage among many trusted servers spared in the network instead of storing in central server. The DSS uses erasure codes for this purpose. The implemented system consists of one Primary Server (PS) that has the main database, two Storage Servers (SS) responsible for storing patient data, and one Reader Server (RS) that collects patient information from SSs to read it. The PS encrypts the data using Redundant Residue Number System (RRNS) technique that is depending on a library of moduli in the encrypting process to generate residues. Three algorithms have been implemented to decode RRNS: Chinese Remainder Theorem (CRT), base extension (BEX) with mixed radix conversion, and New Chinese Remainder Theorems (CRT I). RSA has been used to encrypt any exchange messages between servers, and DSA has been used for signing the residues and File Descriptor (FDs) before sending process.

After implementing of the system, simulation results have shown that whenever the size of moduli is large, the time of the encoding and decoding process and the size of the sent message will be small and the code efficiency is high. CRT I has considered to be the best for decoding RRNS because it makes the computations faster and functional with low overheads. The security of the system has been ensured where RRNS provide confidentiality and dependability services for the patient data, in addition to using DSA for ensuring integrity and RSA for archiving confidentiality, authentication, and non-repudiation.

Abbreviations

AES	Advanced Encryption Standard
ALTR	Adaptive Least Temperature Routing
ACL	Access Control list
ASCII	American Standard Code for Information Interchange
BAN	Body area network
BSN	Body Sensor Networks
BCU	Body Control Unit
BER	Bit Error Rate
BP	Blood Pressure
BEX	Base Extension
CSMA/CA	Carrier Sense Multiple Access/ Collision Avoidance
CCA	Clear Channel Assessment
CAP	Contention Access Phase
CRT	Chinese Remainder Theorem
CRT I	New Chinese Remainder Theorems
CE	Code Efficiency
CC	Cyclomatic Complexity
DSA	Digital Signature Algorithm
DB	Data Base
EEG	Electroencephalography
ECG	Electrocardiography
EFC	Electrostatic Field Communication
EAP1	Exclusive Access Phase
ECC	Elliptic curve cryptography
FD	File Descriptor
GPRS	General Packet Radio Service
GPS	Global Position System
GUI	Graphical User Interface
HBC	Human Body Communications PHY
HIT	Hybrid Indirect Transmissions
ISM	Industrial, Scientific and Medical
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IDs	Identifier
LAN	Local Area Network
LEACH	Low Energy Adaptive Clustering Hierarchy
LRC	Locally Repairable Codes

LTR	Least Temperature Routing
MAN	Metropolitan Area Network
MANET	Mobile Ad-hoc Networks
MICS	Medical Implant Communications Service
MAC	Medium Access Control
MAC	Message Authentication Codes
MRC	Mixed Radix Conversion
MRD	Mixed Radix Digits
MI	Multiplicative Inverse
MS	Medical server
NB	Narrowband
PAN	Personal Area Network
PDA	Personal Digital Assistants
PDR	Progressive Data Retrieval
PHY	Physical Layer
PN	Patient Number
SPINS	Security Protocols for Sensor Networks
PS	Primary Server
QoS	Quality of Service
RAP1	Random Access Phase
RNS	Residue Number System
RRNS	Redundant Residue Number System
RSA	Rivest-Shamir-Adleman
RS	Reader Server
SN	Sensor Nodes
SPN	Single Point of Failure
SS	Storage Server
TEG	Thermo- Electric Generator
TDMA	Time Division Multiple Access
TARA	Thermal Aware Routing Algorithm
UWB	Ultra-Wideband
WAN	Wide Area Network
WSN	Wireless Sensor Network
WMN	Wireless Mesh Networks
WLAN	Wireless Local Area Networks
WPAN	Wireless Personal Area Network
WMAN	Wireless Metropolitan Area Networks
WBAN	Wireless Body Area Networks
WMTS	Wireless Medical Telemetry Services

Contents

Chapter Three: Distributed data security and privacy in WBANs

Chapter Four: Proposed System design and Implementation

Chapter Five: Results and Discussion

List of Figures

List of Tables

Chapter One

General Introduction

Chapter one

General Introduction

1.1. Introduction

Network is defined as two or more computers that are connected to support many types of the sharing such as sharing resources like files or printers or sharing service such as an internet connection. Figure 1.1 explains the concept of network generally [1].

The connected computers must be linked through path-way called" ***transmission medium***" such as cables, telephone line; radio waves or infrared light beams, etc. The set of communication rules called "***protocols***" must be followed by computers on the network; these protocols are used for arriving data at its intended destination and also used for sending and receiving systems to understand each other [2].

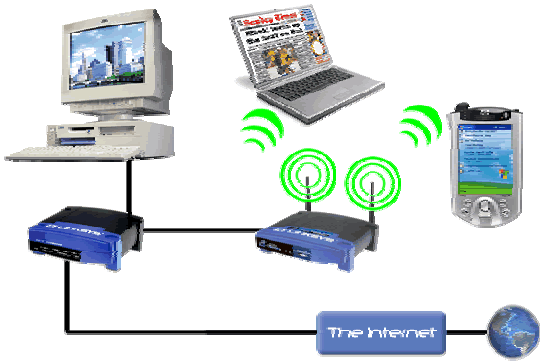


Figure 1.1: General Networking Architecture

There are different types of networks that are divided according to many criteria. For example, according to ***access restriction*** networks can be divided into **private** and **public** networks. Example of the private networks is networks maintained by banks, while public networks are generally accessible to the average user, but they may require registration and payment of connection fees. The internet is an example of a public network. Also the network can be classified according to ***Communication model employed by the nodes*** in the network communicating based on a **point-to-point** or

broadcast models. In the point-to-point model, a message follows dedicated route through the network from one node to another, while in the broadcast model the medium is shared by multiple nodes and the message sent by a node is received by all others. The address in the message determines in which node the message is intended [3].

An important taxonomy of network classification is according to the ***Geographic spread of nodes and hosts*** such as Body Area network (**BAN**), Personal Area Network (**PAN**), Local Area Network (**LAN**), Metropolitan Area Network (**MAN**) or Wide Area Network (**WAN**) as shown in Figure 1.2. The reason for such classification is that the size of a network often has implications for the underlying technology that can be used, with a key factor being the amount of time it takes for data to propagate from one end of the network to the other [4].

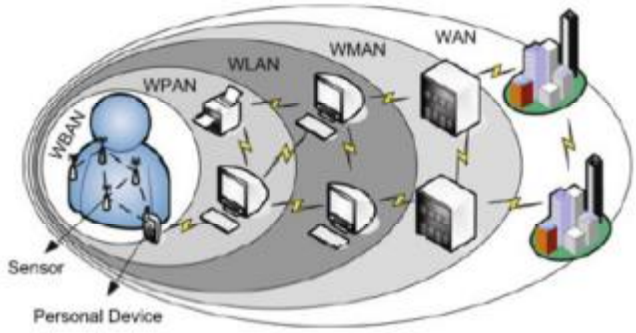


Figure 1.2: Types of Wireless Networks [5]

1.2. Wireless Networks

In general wireless networking can be categorized into two classes: ***cellular*** *and* ***wireless ad hoc*** networking. The difference between these classes is the existence of a fixed infrastructure [6]. The wireless users keep connected with the wireless system when they roam from one place to another by a centralized supporting structure called an ***access point*** [7]. In the cellular mobile system there are many access points connected to each other using a

fixed infrastructure; terminals in this system reach an access point by using a single-hop wireless link [6].

1.2.1. Mobile Ad-hoc Networks (MANT)

MANETs defines a new kind of wireless systems operating in the absence of a fixed infrastructure [7], and it consists of a collection of mobile users communicating over a relatively bandwidth-constrained wireless link with limited battery power. Because of the mobility in the network, topology is dynamic and may change rapidly over time. The node in a wireless ad hoc network can perform many functions such as a transmitter, host, and a router; the reason for this is to maintain communication between the nodes in the network. MANET needs efficient distributed algorithms for the purpose of determining network organization, link scheduling, power control, and routing [8]. The system is also called ***"multi-hop wireless ad hoc networks"*** because the route between nodes may include multiple hops. If two nodes reside in the same transmission range, they can communicate directly "peer-to-peer". Otherwise, the nodes use intermediate nodes for doing the communication hop by hop.

Wireless sensor networks take a special role in the ad hoc networking field[7]. Figure 1.3 shows the general form of Mobile Ad-hoc Networks. The most popular applications of MANT are: ***Health care*** (where it used to monitor the health conditions and whereabouts of patients and elderly people), ***Entertainment*** (such as Multi-user games, robotic pets) and used in ***Disaster relief operations*** (management of relief operations after large-scale disasters such as earthquakes, tsunamis and floods, because the capability of rapid deployment that makes them an prominent technology) [6].

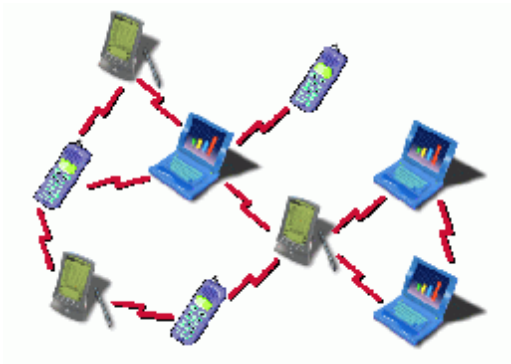


Figure 1.3: Mobile Ad-hoc Networks

1.2.2. Wireless sensor network (WSNs)

Wireless sensor network (WSN) is defined as a network represented by nodes that can sense the environment and transmit the information gathered from the monitored field through wireless links. WSN has severe limitations in battery power, size and device capabilities [9]. Each node in WSN has many components as follows: processor (microcontrollers or CPUs), memory (program, data and flash memories), RF transceiver (usually with a single Omni- directional antenna), a power source (e.g., batteries and solar cells), and accommodate various sensors and actuators [10]. The sensor takes different forms such as camera, thermal, microphone or other; the sensors can be used for different purposes such as temperature, humidity, vehicular movement or monitor the pressure [11]. Wireless sensor networks reduce the installation costs and can adapt to change in the environments. Wireless sensor networks do not rely on any pre-existing infrastructure. When any node in the WSN fails, the network should continue to deliver date by selecting new topology [12]. The cluster based hierarchical model may be the most efficient and adaptive routing model in the sensor network. The cluster formation is an important factor in the cost reduction. The "cost" here refers to the spending of setup and maintenance of the sensor networks [13]. WSNs are used in many applications such as ***Military applications*** because it has many useful features like rapid deployment, self-organization, fault tolerance and security [11]. They are also used in ***Environmental applications*** such as

tracking the movements of birds, small animals, forest fire detection, precision agriculture etc. The most important application can be in ***Health system*****,** where they can provide interfaces for the disabled; integrated monitoring for the patient; diagnostics; drug administration, and finally tracking, monitoring doctors and patients in a hospital [14]. A simple form of WSN is shown in Figure 1.4.

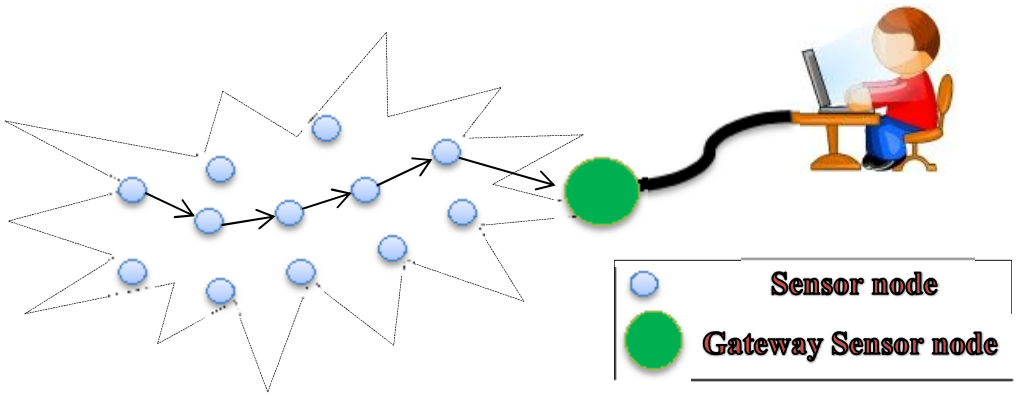


Figure 1.4: Wireless sensor network

1.2.3. Wireless Mesh Networks (WMNs)

The definition of a wireless mesh network is a packet-switched network with a constant wireless backbone. The modifications on infrastructure happen only when add/remove or failure of access points [15]. Wireless mesh networks (WMNs) can be composed of two types of nodes; fixed and mobile nodes that are connected by wireless links to form a multi-hop ad hoc network[16]. The nodes in WMNs operate as a host and as a router delivering packets to the destinations when the sender is outside the wireless transmission range.

Generally WMNs consist of mesh routers and mesh clients, the mesh routers are low mobility while mesh clients can be either fixed or mobile [17]. A mesh router usually contains at least two radios, one to communicate with mesh clients and others to pass data to other mesh routers and also contain various network interfaces such as IEEE 802.3, IEEE 802.11, IEEE 802.15 and IEEE 802.16. Mesh routers can be classified as *access*, *backbone* or

gateway mesh routers [6]. WMNs may resolve the limitations and improve the performance of many networks like ad hoc networks, wireless local area networks (WLANs), wireless personal area networks (WPANs) and wireless metropolitan area networks (WMANs) [17]. The applications implemented using an ad hoc network can also implement by WMNs.

1.3. Background of Wireless Body Area Network

Wireless body area networks can be considered as a mix among the above three wireless networks. It is an emerging technology; therefore it has a short history. Many factors have led to interest in health care in many countries such as the increase of average lifespan and health cost. The advances in miniaturization of electronic devices, sensing, battery and wireless communication technologies have considered the main factors with these above which led to the growth of Wireless Body Area Networks (WBANs)[18][19].

Van Dam et al, in 2001 is coined the term "Wireless Body Area Network" (WBAN) [5] which has several other names such as Body Area Network (BAN), and Body Sensor Networks (BSN). The reason for the popularity of WBAN is that patient data monitoring is considered to be a main issue for health & disease management [20].

WBAN includes sensors that are either implanted or wearable in or on the human body that monitor the state of health during normal daily activities for long periods of time [21]. The second part of this network is a personal device that collects the monitored signals. PDAs or smart phones are example of such a device that transmits the data to a healthcare professional residing in the hospital for health monitoring [22]. Without using such networks the number of the patient on health care systems will increase such that it makes the introduced service quality would be lower and cause letdown of medical staff, errors, and sometimes deaths [23].

Using a wired connection for these systems makes the network so heavy and costly for deployment and maintenance [5]. Also using wires to connect sensors on the patient is very uncomfortable. The benefit of using wireless body area networks is that the doctor can monitor the patient remotely and thus reduce the load on hospital [20]. The wireless technology makes the cost effective and patient comfortable in movement and there is no need to keep him/her in the hospital.

Health care jobs that are supported by electronic processes and communications are called e-Health. The health care is in progress to become mobile then it is defined as m-Health [5]. Nowadays WBANs are becoming important in addition to healthcare systems, for sporting activities, and members of emergency and military services [21]. The WBAN system is illustrated in Figure 1.5 bellow.

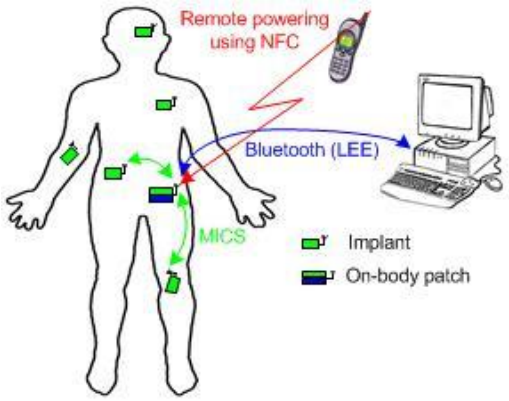


Figure 1.5: Wireless Body Area Networks

1.4. E-Health system

The prefix "e-", refer to "electronic", where many applications also use this term such as "e-learning", "e-governance" and "e-transport", in order to represent the notion of digital data. The results that are obtained without digitization are no automatic processing and no immediate exchange via the network [24].

E-health is defined as an approach used to improve the access, efficiency, effectiveness, and quality of clinical and business processes exercised by

healthcare organizations, patients, and consumers. These improvements are achieved by the application of Internet and other related technologies in the healthcare industry [25].

1.4.1. The domain of e-Health system

There are three major domains of an e-Health system which include mobile healthcare, remote healthcare, and e-hospital as shown in Figure 1.6. The definition of each domain is as follows [26]:

1) ***Mobile Healthcare***: It uses the mobile devices such as cell phones and personal digital assistants (PDA) for healthcare service to allow users to manage anytime and anywhere they want their health aspects.
2) ***Remote health care:*** The patient is diagnosed even if he/she is away from the hospital by using remote healthcare that uses the internet or other wireless technology to access to patient
3) ***E-Hospital:*** By applying information and communication technology (ICT) in the in-hospital, hospital-to- hospital, and hospital-to-pharmacy management procedures then e-Hospital service will give optimal efficiency in hospital management system such as prescription delivery, patient booking, and diagnosis.

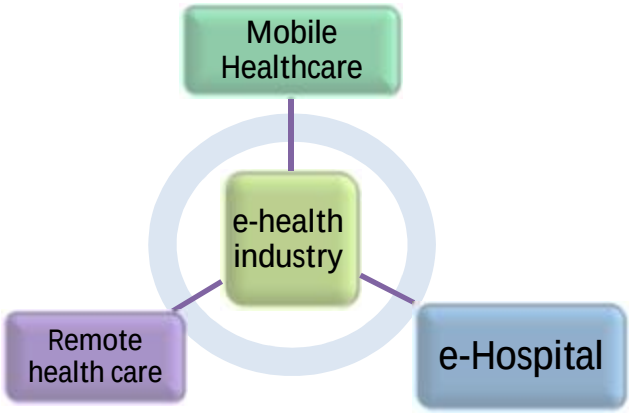


Figure 1.6: Scope of e-Health industry

1.4.2. The goals of e-health

Since e-health uses the internet, it tries to achieve many goals beyond this feature. Some of these goals of e-health include [27]:

1) ***Efficiency:*** Increasing efficiency in healthcare is one of the promises of e-health; this goal can be achieved by the help of the internet. As a result of increasing the efficiency, the cost is decreasing. The possible ways for decreasing cost are avoiding duplicate or unnecessary diagnostic or curative interventions.
2) ***Empowerment of consumers and patients***: E-health systems open new ways for a patient-center medicine, enables patient education, and do this by making the knowledge bases of medicine and personal electronic records attainable to consumers over the internet.
3) ***Ethics:*** This feature is not dedicated to e-health system alone but it is a feature of the internet technology. E-health encompasses new styles of the interaction between the patient and the physician and consequently will impose new challenges and threats to ethical issues such as online professional practice, privacy and security issues and others.
4) ***Education of physicians:*** The physicians can obtain the education through online sources (continuing medical education) and also the consumers can take the education through (health education, tailored preventive information for consumers). This education of physicians and consumers, remain them up to date with the latest developments in the medical areas of their respective interests.

1.5. Literature Survey

In this section a survey of some significant related works that deal with a data storage system in WBANs are presented:

1) ***M. O. Rabin (1989)*** proposed a technique for fault-tolerant file server using an Information Dispersal Algorithm (IDA) that breaks a file into *n* pieces so that every *m* pieces suffice for reconstructing file. IDA disperses the information on file into *n* pieces or locations in a reliable way. The main advantage of this dispersal method is space efficiency. Both of the dispersal and reconstruction are computationally efficient. The main disadvantage of IDA is that it did not address the confidentiality issues [28].

2) ***Chessa and P. Maestrini (2003)*** proposed a method for distributed storage by using RRNS to encode the file. This system achieves the Dependability where the data can be reconstructed until $s \leq r$ residue erasures, combined with up to $(r-s)/2$ corrupted residues. It also ensures data confidentiality since recovering the original information requires knowledge of the all modules (keys). However, this scheme lacks authentication and integrity capabilities [29].

3) ***R. Ball, J. Grant, J. So, V. Spurrett, and R. Lemos (2007)*** proposed fragmentation redundancy-scattering (FRS) technique that is fragmented the confidential information into insignificant fragments, and scattered these fragments in a redundant fashion through a network. Two algorithms are developed to maintain a constant number of fragment replicas: one based on the game of life, and the other based on roaming ants. This paper researched the use of autonomous agents combined with an intrusion tolerance technique for providing secure and dependable storage for ad hoc networks [30].

4) ***Q. Wang, K. Ren, W. Lou and Y. Zhang (2009)*** proposed a scheme to achieve security, dependability, and dynamic integrity in data storage. In the initial data storage process, they utilized perfect secret sharing and erasure coding to guarantee data confidentiality and dependability. Based on the principle of algebraic signatures, they constructed efficient dynamic data integrity checking scheme to ensure the integrity of data shares. A weak point of this scheme is that it does not allow a third party to perform integrity checks. This is quite unsuitable in WBAN applications because we want the local server to verify the integrity of the collected data [31].

5) ***Y. S. Han, S. Omiwade, and R. Zheng (2012)*** designed progressive data retrieval PDR algorithm for distributed storage systems that is highly computation-efficient and communication-optimal algorithm. The communication and computation costs for data retrieval are minimized by utilizing intermediate computation results and retrieving only the minimum data required for successful data reconstruction, respectively. The proposal shows that decentralized fountain codes and PDR for distributed storage systems are most suitable to networks without Byzantine storage nodes because of the associated minimal computation cost for fountain codes, and the minimal data retrieval cost of PDR [32].

6) ***J. Pääkkönen (2012)*** focused on the designing a distributed storage system that takes advantage of the available storage capacity of mobile terminals in order to decrease the expected power consumption of wireless transmission systems. This paper is used regenerating codes to add a redundant block of encoded data, in order to provide a file with redundancy (Regenerating codes are erasure codes that are specifically designed for distributed storage.). This method was the most energy

efficient solution. Regenerating codes appear to be the best method for systems that suffer from multiple storage node failures [33].

7) ***T. Chareonvisal (2012)*** introduced Network coding to improve distributed storage system: It suggested the way to improve distributed storage system such as increase a chance to recover data in case there is a fail storage node or link fails in a network. It studied different schemes of a distributed storage system such as replication code, the maximum distance separable and regenerating code and it explained all aspects of these methods [34].

8) ***M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. Dimakis, R. Vadali, S. Chen and D. Borthakur (2013)*** introduced a new family of erasure codes called Locally Repairable Codes (LRCs) that is creating a new operating point that will be practically relevant in large-scale storage systems, especially when the network bandwidth is the main performance bottleneck. Deploying large LRCs offer high fault tolerance and small storage overhead [35].

9) ***L. Juarez, F. Oggier and A. Datta (2013)*** proposed a new decentralized erasure coding that reduced the network traffic required to archive replicated data in distributed storage systems. The proposed approach exploits the presence of data that is already replicated through the system and distributes the redundancy generation among those nodes that store part of this replicated data, which in turn reduces the total amount of data transferred during the encoding process. By storing additional replicated blocks at nodes executing the distributed encoding tasks, the necessary network traffic for archiving can be further reduced[36].

1.6. Work Objectives

The main aims of this thesis are:

1) Investigating the use of a distributed storage system for storing patient information located in the top tier of WBAN in a secure manner, this tier has the database that is found in the hospital.

2) Achieving confidentiality, dependability in the Distributed storage system by using Redundant Residue Number System (RRNS).

3) Achieving authenticity, non-repudiation and dynamic integrity in that storage system by using public-key based cryptography and digital signature techniques.

4) Using three of the best algorithms for Decoding Redundant Residue Number Systems and comparing between them.

1.7. Thesis Layout

- ***Chapter two:*** In chapter two more details about wireless body area networks are given, such as their architecture, characteristics, challenges and applications.
- ***Chapter three:*** In this chapter central and distributed storage systems are explained and compared to each other in order to show the security aspects of distributed storage systems.
- ***Chapter four:*** Chapter four is dedicated to present the proposed system for achieving security and privacy of a distributed storage system. This includes various design aspects, interface, and algorithms used.
- ***Chapter five:*** In this chapter, the software implementation details and simulation results of the prototype system are present and its GUI is illustrated, based on MS Windows Visual C# environments.
- ***Chapter six:*** This chapter contains the most important conclusions and some future works that are suggests for developing our work.

Chapter Two

State-of-the-Art in Wireless Body Area Networks

Chapter two
State-of-the-Art in Wireless Body Area Networks

2.1. Introduction

WBAN is defined as a special purpose sensor network designed to turn on autonomously to connect different small and intelligent devices (nodes) that existing inside and outside of a human body [37]. These nodes are portable, miniaturized, and autonomous that monitors the body function for sporting, health, entertainment, and emergency applications [38]. Vital signs such as (heart rate, blood pressure, oxygen saturation, activity) or environmental parameters like (location, temperature, and humidity, light) can be sampled, processed, and communicated by these devices [39].

In order to store collected data, the devices can search and find a suitable communication network and transmit data to a remote database server resides in the hospital or to medical centers where the professionals analyze this data[37]. The communication between sensors in a WBAN and servers through the Internet is private and confidential; therefore it must be encrypted in order to protect the patient's privacy [40].

In this chapter WBANs are considered in more details, including their structure, characteristics and challenges. The most important requirements of such network, type of the devices used in the health care system, and how WBANs are employed in real life are shown in this chapter. At the end of this chapter an overview of the protocols and routing in the WBAN are given.

2.2. E-Health System architecture

The patient’s vital information can be monitored continuously for diagnosis and prescription by using in-body and on-body nodes. WBANs used many bands for data transmission such as Wireless Medical Telemetry Services

(***WMTS***) this is the licensed band. The feature of fewer interfering sources for this band made it used for medical applications where only authorized users such as physicians and trained technicians can use this band. WMTS band cannot support video and voice transmissions because it has (14 MHz) bandwidth, therefor another band is used for medical applications named industrial, scientific and medical (***ISM***) band with (2.4 GHz) this band can protect adjacent channel interference because it includes guard bands. For the implant communication licensed Medical Implant Communications Service (MICS) bands (402-405 MHz) is utilized. Ultra-Wideband (***UWB***) also is used in the WBANs communication with (3.1-10.6GHz) [38] [41].

2.2.1. WBAN Traffic Categories

The WBAN traffic is divided into three categories that can be for medical and nonmedical applications:

- ***On demand***
- ***Emergency***
- ***Normal traffic***

On-demand traffic is initiated by the doctor in order to obtain certain information for diagnostic purposes. The kind of traffic that initiated with nodes when they pass a predefined threshold called ***Emergency traffic*** and these nodes should be accommodated immediately, while the normal condition of the patients is monitored without any criticality and on demand events the ***Normal traffic*** is used. This contains hidden & routine health monitoring of patients, and treatment of many diseases [20].

2.2.2. The architecture of WBAN

The broader multi-tier telemedicine system is the architecture of WBAN that are proposed for health monitoring. Medical server (MS) is the top tier. Hundreds or thousands of individual users are served by the medical server.

This server encompasses a complex network of interconnected services, medical personnel, and healthcare professionals. The functions of medical server are: keeping electronic medical records of registered users and providing various services to the users and medical personnel.

The second tier of the architecture is the personal server or Gateway node, personal server has many functions like interfaces WBAN sensor nodes, provides the graphical user interface, and communicates with services at the top tier. Personal digital assistant (PDAs) or a cell phone used to implement personal server [39]. The connection with other networks is the responsibility of a PDA or smart phone. For example forward obtained data to the hospital server, therefore an important challenging task in WBANS is forwarding data. Routing approaches are used to overcome this challenge [42]. The personal server employs mobile telephone networks (2G, GPRS and 3G) or WLANs to reach an internet access point; consequently it can communicate with the medical server.

Tier 1 considered a pivotal part of the telemedical system that is consisting of a number of intelligent sensor nodes (SN). Each sensor receives initialization commands and responds to queries from the personal server. These sensor nodes are placed on the user's body. The main functions of these sensor nodes are: sample vital signs unobtrusively and transfer the relevant data to a personal server, doing this by using wireless personal network implemented using ZigBee (802.15.4) or Bluetooth (802.15.1). Via the Internet the patient’s physician can access to the data from his/her office and can perform many functions such as ensure that the patient is within the expected health (heart rate, blood pressure), ensure that the patient is responding to a given treatment or that a patient has been performing the given exercises. Figure 2.1 shows the architecture of WBAN. The personal server application should determine the user’s state and his or her health status, based on synergy of information from multiple medical sensors, and

consequently provides feedback through a user friendly and intuitive graphical or audio user interface. The personal server establishes a secure communication to the medical server if the communication channel to the medical server is available and it will sends reports that can be integrated into the user's medical record. While the personal server should be able to store the data locally if a link between the personal server and the medical server is not available and it will initiate data uploads when a link becomes available[37] [39].

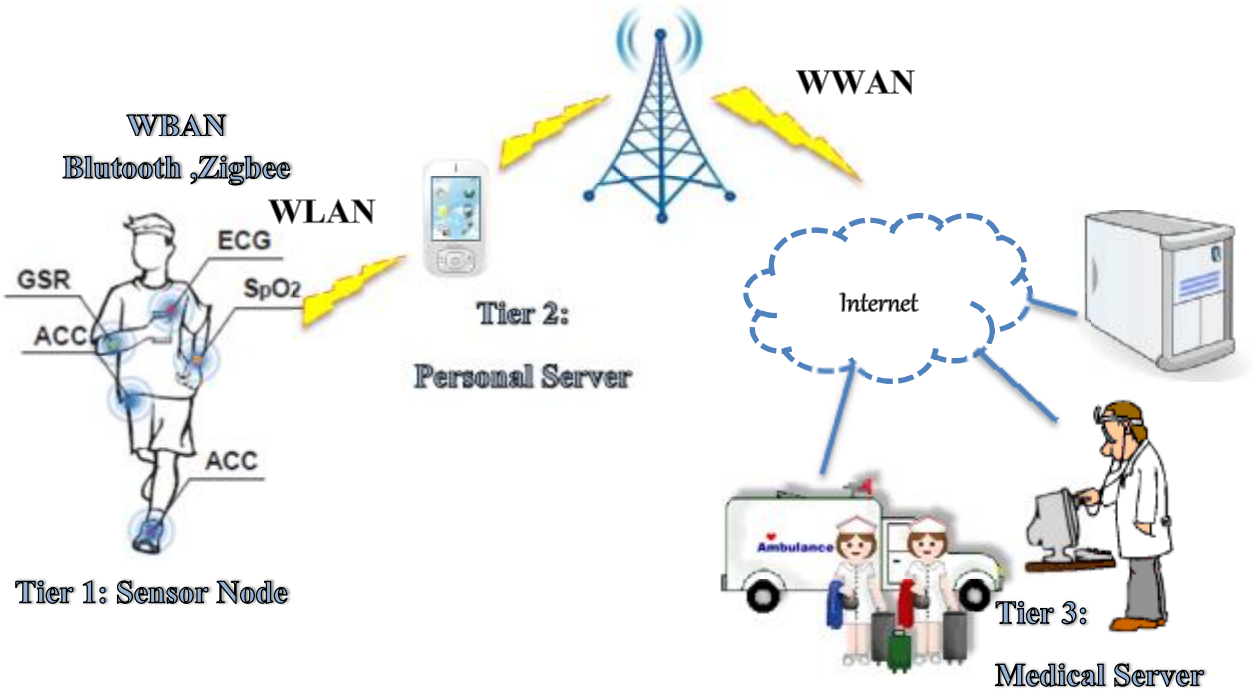


Figure 2.1: General Multi-tier Telemedicine Architecture

2.3. Challenges and Characteristics of WBANs

The challenges appear in WBAN are as following [22] [43]:

1) The scale of WBAN is limited to the human body (cm/m)
2) The node numbers in this network are a few, limited in space. It has a less than 50 nodes.
3) In this technology, node performs multiple tasks

4) The size of the node must be small because it may implant in the human body.
5) The topology of this network is more variable due to body movement therefor WBANs should be robust against frequent topology changes.
6) The process of replacement node is difficult if the nodes are implanted.
7) Although a long lifetime of the device like several years/months especially for implanted devices is required and it is impossible to recharge or change the batteries for most devices. Hence, the energy resources, the computational power and available memory of such devices must be limited;
8) Energy scavenging : the source of energy scavenges most likely to be motion (vibration) and thermal (body heat)
9) To make the patient information strictly private and confidential, the security level must be higher and stringent security mechanisms are needed.
10) The impact of loss the data is more significant and may require extra measures to ensure real-time data delivery and QoS.
11) High reliability and low delay is required because we are dealing with data often consists of medical information.

The unique characteristics of WBANs compared to other networks are as follows [5] [22]:

1) WBANs are considered as small-scale networks, and it has a short communication range including the communication with or on a human body.
2) The devices in WBAN are very limited in their computational capabilities, power especially if is implanted into the body, and required scalable performances; data rate up to 10Mbps, peak power consumption up to 40MW.

3) To minimize interference and to cope with health concerns; extremely low transmit power per node is needed.
4) The nature of the data that are detected, collected and transmitted in WBAN is relatively sensitive; hence it demands high security and privacy.
5) The topology of WBAN is "star topology", where communication centrally organized and every sensor node is directly linked to a master node. This topology cannot always satisfy the desired reliability requirement therefore a star-mesh hybrid topology extends the conventional approach and creates mesh networking among central coordinators in multiple star networks.

2.4. Requirements of WBANs

There are important different aspects of the system requirements of this network such as type of devices, data rate and energy and others [5] [22] [43]. Overview of each one is given.

2.4.1 Types of devices

There are four devices that can be found in WBAN systems:

a) Sensor node **(*Wireless*):** Sensor device consists of several components such as sensor hardware, a power unit, a processor, memory, and a transmitter or transceiver. This device performs many functions like responding to and gathering data on physical stimuli, processes the data if necessary, and reports this information wirelessly.

b) Actuator node (Wireless): Actuators have the same ingredients found in the sensor device that are, actuator hardware (e.g. Hardware for medicine administration), a power unit, a processor, memory, and a receiver or transceiver. The purpose of this device relies on the data obtained from the sensors or through interplay with the user

c) Gateway (Personal Device): This device consists of a power unit, memory, and a transceiver. Gathers all the data acquired by the sensor nodes and informs the users, this function of such device. Another name of this device is a Body Control Unit (BCU), body gateway, or a sink

***d) Monitoring Server*:** The components of this type of equipment are database for data storage, processing, and analyzing software for delivering system intended services.

2.4.2 Data rates

Data rates multifarious from a few k bit/s of simple data to several Mbit/s at video streams this multifarious belong to the strong heterogeneity of the applications. Data can be sent in higher rate if it sent in bursts. Bit error rate (BER) is a term used to quantify the number of lost packets consequently; it used to provide the reliability of the data transmission. The reliability depends upon the data rate in the medical device; the devices with a higher data rate require a lower BER, while Low data rate devices require a high BER [5][22].

2.4.3 Energy

Energy consumption in the WBAN technology divided into three areas: sensing, communication, and data processing. Often the most power consumption occurs in the wireless communication. The node power is often restricted. Batteries are kept small and the energy consumption of the devices is reduced, the reason for this is that the largest contributor to the sensor device in terms of both dimensions and weight is the size of the battery that are used to store the needed energy.

The lifetime of the battery must be sustained for long period exceeded 5 years in any application (pacemaker or a glucose monitor) without intervention, this supporting occur while the sensors and actuator in WBANs are operating. A Thermo- Electric Generator (TEG) is used to transform the

temperature difference between the environment and the human body into electrical energy, this technique is called "scavenging" that used to enhance the lifetime of a node during the operation of the system. Some systems could run forever if the scavenged energy is larger than the average consumed energy. The ideal solution for accomplishment autonomous WBANs is a combination of lower energy consumption and energy scavenging. The body heat and body vibration on-body sources seem very suitable energy scavenging [5]. There are other techniques may be used for power saving such as avoid unnecessary retransmissions, minimize idle listening or overhearing by the put receiver in standby mode or sleep mode, and use power efficient error correction schemes [43].

2.5. Sensor types

Sensor nodes can be categorized into wearable and implantable sensors. Some examples for each one are given below [19]:

1) Wearable Sensors

a) Pulse oximetry: The oxygen saturation levels (SpO2) in a blood and the mutations in blood volume in the skin that sync with the cardiac cycle can be measured indirectly by using a medical device called a pulse oximeter. This device attaches to a finger or an earlobe.

b) Electrocardiography (ECG): ECG waveform is used to represent the propagation of electric potentials through the heart muscle with respect to time.

c) Blood Pressure: A blood pressure (BP) used to measure the force made by circulating blood on the walls of blood vessels. BP takes two cases during a cardiac cycle, a maximum (systolic) and a minimum (diastolic) pressure. Figure 2.2 shows BP device.

Figure 2.2: Blood Pressure (BP)

d) Electroencephalography (EEG): The electrical activity of the brain can be represented by EEG. Ambulatory EEG (AEEG) can be used for diagnosis of the epilepsy disease and in the monitoring of patient response to treatment.

2) Implantable Sensors

a) Glucose Monitoring: The glucose levels can be monitored by placing an implantable sensor covered with a multilayered membrane in the subcutaneous callus of the tummy as shown in Figure 2.3. The glucose levels are determined every 30 s and every 5 minutes the radio transmission of the glucose data occurs.

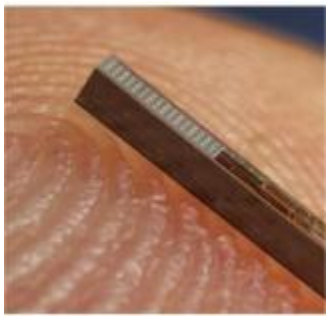


Figure 2.3: Implantable Sensors

2.6. WBAN Applications

The applications of WBANs can be divided into medical and non-medical applications, some of these are explained below. "*WBANs improve the user's Quality of Life*" this is the main feature of all these applications [5].

2.6.1. Medical applications

The WBAN technology involves collecting vital information about a patient and forwards it to a remote station for analysis. The huge amount of information can be used to treat various diseases; we will show some of these as follows [44]:

1) ***Cancer Detection***: One of the largest threats to the human life is cancer disease. The physician can diagnose the tumors without biopsy by using a set of smaller sensors capable of monitoring cancer cells.
2) ***Asthma:*** By monitoring allergic agents in the air and by providing real-time feedback to the physician, a WBAN can help a large number of patients suffering from asthma disease. Chu et al proposed a GPS (Global Position System) device that monitors environmental factors and send an alarm to the patient when it detects information allergic[38].
3) ***Diabetes:*** A WBAN can be helpful in the case of diabetes, where proper dosing and reduces the risk of fainting and the blindness and other complications can be done by frequent monitoring [43].
4) ***Artificial Retina:*** This retina containing a matrix of micro sensors that can be implanted in the eye below the surface of the retina. The electrical impulses translated into neurological signals .The input can be obtained locally from light sensitive sensors or by an external camera mounted on a pair of glasses [44] [5].
5) ***Disabled:*** A WBAN can also be used to offer assistance to the disabled, sensors and actuators in this application are used where a paraplegic can be equipped with sensors deciding the position of the legs or with sensors attached to the nerves, the actuators positioned on the legs can stimulate the muscles. The possibility to move can be done by the interaction between the data from the sensors and the actuators [5].

2.6.2. Non-medical applications

These include:

1) ***Battlefield:*** To connect soldiers in a battlefield and report their actions to the commander, WBANs can be applied. In order to avoid ambushes, the soldiers should have a secure communication channel[38].
2) ***Elderly people:*** By using sensors, WBAN technology can be used to monitor the health status of the elderly people e.g. a fall consequently the elder can live longer in their own home.
3) ***Harmful toxins*** in for example the food ingested and air inhaled can detect by placing sensors in the nasal area or the tongue [43].

2.7. Network Protocols for WBANs

The communication range of WBAN is restricted within a few meters because this network operating close to the human body. The IEEE defined enabling technology for each type of network such as a WPAN uses IEEE 802.15.1 (Bluetooth) or IEEE 802.15.4 (ZigBee), WLAN uses the IEEE 802.11 (Wi-Fi) and WMAN uses IEEE 802.16 (WiMax). The communication in WAN can be established via satellite links. The IEEE has set up a working group to elaborate a standard for communication in a WBAN [43].

2.7.1. Medium Access Control (MAC) layer

In a wireless environment, the MAC-protocol is responsible for tolerance overhearing, idle listening and collisions. It considered a critical factor in a WBAN because it strongly influences the energy consumption of the nodes in the network [43]. There are limited numbers of MAC-protocols specially developed for WBANs [5]. MAC-protocols are designed for wireless sensor networks can be categorized into ***contention-based*** and ***schedule-based*** [43]. An example of the contention-based is CSMA/CA while TDMA is a typical scheme for schedule-based.

Contention-based approaches have some advantages like the simplicity, its infrastructure-free ad hoc feature and good adaptability to traffic fluctuation, especially for low load. Controlling the power and duty cycle of the radio techniques commonly reduces energy consumption in contention-based protocols. On the other hand schedule-based approaches are free of idle listening, overhearing and packet collisions because of the lack of medium competition, but require tight time synchronization [5].

Bluetooth (IEEE 802.15.1) is used in some implementations of WBANs, but this was developed as a cable replacement and does not support (or only very limited) multi-hop communication, and because it has a complex protocol stack and high energy consumption compared to IEEE 802.15.4, it is not suited to be used in a WBAN.

WBANs is used as an alternative solution IEEE 802.15.4, or ZigBee as enabling technology [5] that can be utilized to handle time critical events but they expire in case of low traffic load. IEEE 802.15.4 is used within on-body sensor network applications, but if it used within in-body nodes, IEEE 802.15.4 does not achieve the required power level [38].

2.7.2. IEEE 802.15.4

This defined as a low power and low data rate protocol, it offers high reliability. The physical and the medium access layer are defined by this protocol [43]. IEEE 802.15.4 is not designed to support WBANs, some research concludes that the 802.15.4 is not good in term of power consumption and cannot be used as a single solution for all WBAN applications although it can provide QoS [5].

IEEE 802.15.4 uses CSMA/CA. Because the path loss inside human body results in improper Clear Channel Assessment (CCA), CSMA/CA mechanism doesn't provide reliable solutions for in-body nodes. TDMA-based protocols are an alternative solution to use for WBAN. The performance of a TDMA

protocol is analyzed for an on-body sensor network. After the nodes are finished their transmissions, they go into sleep mode. The Electrocardiogram (ECG) node sleeps after 150 seconds [38].

2.7.3. IEEE 802.15.6

The entire channel is divided into superframe structures in IEEE 802.15.6. The meaning of superframe is "each period of transmission". Each superframe is bounded by a beacon period of equal length. The IEEE 802.15.6 standard includes PHY and MAC Layer Specification [44] [45].

1. ***PHY Layer Specification:*** The physical layer supports three different PHYs, i.e., NB, UWB, and HBC.
    a) *Narrowband PHY (NB):* Many functions associated with NB PHY such as activation/deactivation of the radio transceiver, Clear Channel Assessment (CCA) within the current channel, and data transmission/ reception. The range of this band is 402-405 MHz and primarily corresponds with communication with some devices that are implanted into the body.
    b) *Ultra Wideband* PHY *(UWB):* This band is used to communicate between the nodes on the body surface with any external device but also may be used to communicate with nodes within the body. The UWB PHY is operated by two frequency bands low and high bands. Each band is divided into channels; the low band consists of 3 channels (1-3) only. The high band consists of eight channels (4-11). Low and high bands characterized by a bandwidth of 499.2 MHz.
    c) *Human Body Communications PHY (HBC):* HBC is the Electrostatic Field Communication (EFC) specification of PHY, which encompasses the entire protocol for WBAN such as packet structure, intonation, etc. HBC mode can be used to

communicate between nodes on the body surface but also may be used to communicate with external devices. HBC PHY operates on two frequency bands centered at 16 MHz and 27 MHz with the bandwidth of 4 MHz

2. ***MAC layer specifications:*** The IEEE 802.15.6 network operates in one of the following modes.
    a) *Beacon mode with a beacon period superframe boundaries:* Each superframe is divided into Exclusive Access Phase 1 (EAP1), Random Access Phase 1 (RAP1), Type I/II phase, Exclusive Access Phase 2 (EAP2), Random Access Phase 2 (RAP2), Type I /II phase, and a Contention Access Phase (CAP). For important transmissions with higher priority or for emergency communication, EAP1 and EAP2 are used while RAP1, RAP2, and CAP are used for all other traffic. The Type I/II phases are used for (uplink, downlink, blink, and delay blink) allocation intervals.
    b) *Non-beacon mode with superframe boundaries:* the entire superframe in this mode is either Type I or Type II access phase.
    c) *Non-beacon mode without superframe boundaries:* In this mode, the coordinator provides unscheduled Type II polled allocation only.

2.8. Routing in WBANs

The characteristics of the wireless environment made the task of developing efficient routing protocols in WBANs not a simple task. In WSNs the minimal energy consumption is not the important consideration compared to the maximal throughput and minimal routing overhead considerations [43]. There are many characteristics of a WBAN taken into account the in the designing routing protocol: network partitioning due to postural mobility of

the on-body sensors, high propagation loss across the human body, low transmission power of the sensors, and low reliability of end-to-end path from source to sink [19]. The routing strategies in WBANs classified into:

- *Temperature Based Routing.*
- *Cluster Based Routing.*
- *Cross Layer Based Routing.*

2.8.1. Temperature Based Routing

Radiation absorption and heating effects on the human body are the important issues when wireless transmission around and on the body are considered. Temperature routing avoids heat generation, and used traffic control algorithms such as the Thermal Aware Routing Algorithm (TARA), Least Temperature Routing (LTR) and Adaptive LTR (ALTR).

In TARA the route to high temperature area (hot spots) avoided and the data is routed away from these areas. Figure 2.4 shows Thermal Aware Routing TARA where a sender first tries to send packets to a neighboring node that is on the way to the destination and not a hot spot, this neighboring node sends the packets back to the sender again if it was surrounded by hot spots within its communication range. In this case the sender chooses an alternative node that is not a hotspot, the packets winding the hotspots to the destination as shown. The drawbacks of this algorithm are low network lifetime, a high ratio of dropped packets and do not take reliability into account [43] [46] [47].

Alternative algorithm that is considered an improvement of TARA is LTR as shown in Figure 2.5 that is operated by maintaining the recently visited nodes as listed in the packet. LTR could reduce unnecessary hops and loops and it always chooses nodes that have the least temperature [47].

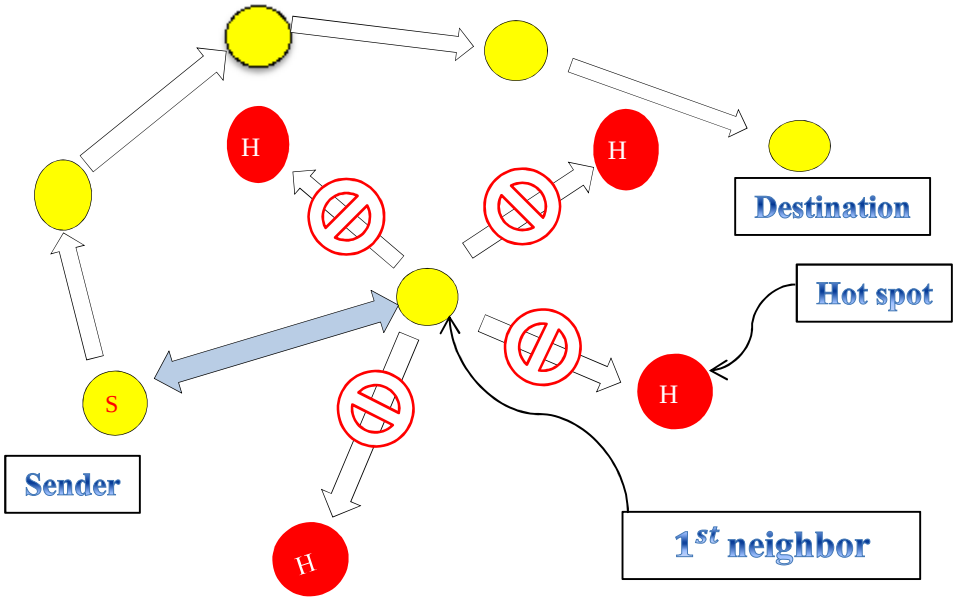


Figure 2.4: Thermal Aware Routing Algorithm (TARA)

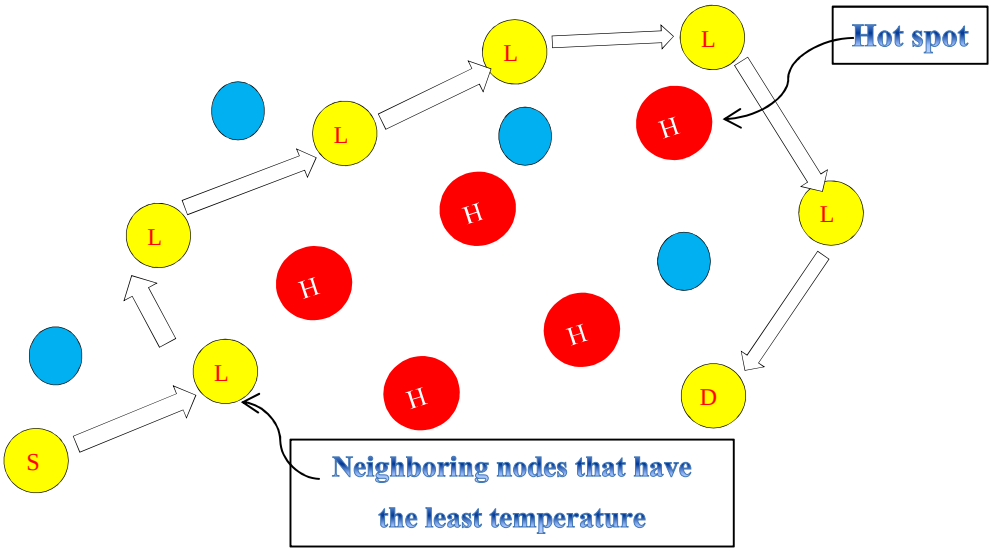


Figure 2.5: Least Temperature Routing

2.8.2. Cluster Based Routing

In order to reduce the number of direct transmissions to the remote base station, Cluster Based Routing is used (Anybody). This protocol is based on Low Energy Adaptive Clustering Hierarchy (LEACH) that randomly selects a cluster head at regular time intervals in order to spread the energy dissipation.

All data are aggregated by the cluster head and sends this data to the base station.

LEACH suffering from drawback, it assumes that all the nodes are within send range of the base station. This problem solved by "Anybody" by changing the cluster head selection and constructing a backbone network of the cluster heads. The energy efficiency is not completely investigated and reliability is not considered [43]. Hybrid Indirect Transmissions (HIT) that combines clustering with forming chains considered Improvement of LEACH[46]. Figure 2.6 shows the cluster based routing.

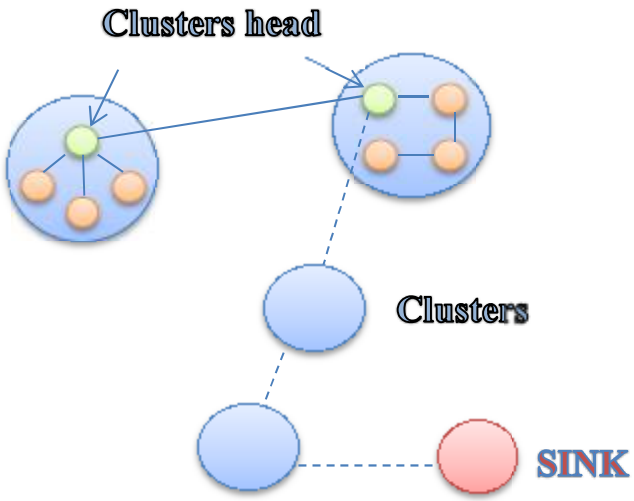


Figure 2.6: Cluster Based Routing

2.8.3. Cross Layer Based Routing

Cross-layer protocols is used as a way to improve the efficiency of and interaction between the protocols in a wireless network by combining two or more layers of the protocol stack [5].

The Cross-Layer Access with Distributed slot Assignment protocol (CICADA) protocol is proposed that sets up a spanning tree and for controlling each node's transmission and reception cycles, this protocol uses time slots [19]. This protocol uses reliable manner to set up a data gathering tree, offering low delay, and high energy efficiency [40].

Of route data from the nodes towards the personal device or sink, CICADA uses data gathering trees that are autonomously set up. In order to lower the interference and avoid idle listening, the time axis is divided into slots. In distributed manner, the slot assignment is done, where each node informs its children when they are allowed to send their data using a SCHEME. The tree is set up in a way that communication is possible between a child and his/her parents or between siblings. It is used a sequence of cycles to define data transfer. Each parent sends a SCHEME-message to all their children, this message contain their slot allocation scheme. Each cycle is divided in two parts: the control sub cycle and the data subcycle. Each subcycle contains slot allocation scheme: the control- scheme and the data-scheme respectively, the control subcycle send these schemes. The control subcycle has ended and the data subcycle starts when all nodes have received their schemes. In the control subcycle, control information is sent downwards from the sink to all nodes. While in the data subcycle, all data is sent upwards to the sink. Data can be sent to the sink in one cycle.

The data scheme consists of three parts: a receiving period, a waiting period, and a sending period. The node must remain silent and should switch off its radio in the waiting period. In the receiving period, the node receives data from its children and in the sending period the node sends data to its parent. To allow new children to join the network, the last slot of each data scheme is a contention slot which is used.

The CICADA protocol does not guarantee any form of security and privacy. Unauthorized nodes can easily join to the WBAN. Suitable security mechanisms have to be added to the CICADA protocol in order to overcome these problems, the result is the CICADA-S, the secure version of the CICADA protocol [40]. Figure 2.7 illustrate the CICADA protocol.

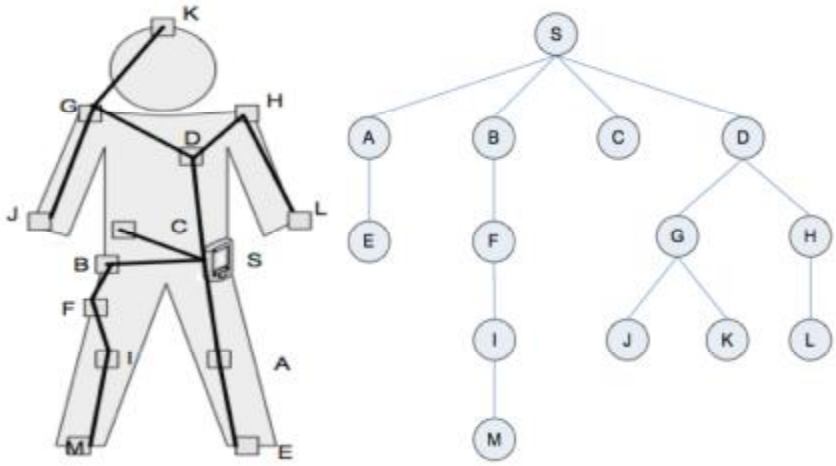


Figure 2.7: CICADA protocol

Chapter Three

Distributed data security and Privacy in WBANs

Chapter Three

Distributed data security and privacy in WBANs

3.1. Introduction

As seen previously, all medical information of patient that are sensitive transferred at least to the back-end server that is managed by the hospital or medical center; therefore this device is considered to be a very critical device. Because of the importance of this part of the network, it is assumed that this server is being located in a secure place in the hospital where it cannot be stolen or tampered and only authorized medical personnel have (partial) access to the server through appropriate identification/authentication mechanisms [40].

Networked storage systems have gained prominence in recent years; this storage has become an efficient way to use in WBANs. This can be divided into two types one based on center point this strategy called Centralized storage, and the other type that does not depend on any center point is called Distributed Storage System [48]. These types of storage systems are described in this chapter in more details, comparison between them, and the advantage and disadvantage are shown in order to decide the batter one to be used in storing server's information. Indeed, types of threats that can compromise the storage server and the security service that must be achieved in this system are discussed.

3.2. Centralized storage

This type of storage is usually found in the Enterprise because it has lower cost of ownership. This type is distinguished by having a central server control of all computing power, processing, program installations, back-ups and file structures, all users purchased and shared the resources of that server[49]**.** The feature of centralized storage is simple but it can lead to a

single point of failure, such as if confidential data is stored on one server and if that server failed (or compromised), the result may be loss or damage of all information. Likewise, the case of network disconnections may prevent any neighborhood servers from accessing data for long periods. Using this type of storage usually cannot provide the level of dependability or availability [29].

3.3. Distributed Storage Systems (DSS)

The definition of distributed storage systems according to Tanenbaum is a "collection of independent computers that appear to the users of the system as a single computer" [50]. In order to achieve a specific goal some resources must be shared such as storage space, computational time or so on. The structure of DSS must tolerance the openness and failure, and must achieve security, scalability, concurrency, and transparency [50]. The storage must be distributed and redundant to ensure all these security and operational requirements [29]. The environment of DSS is ad-hoc network [49] that is consist of a groups of individually unreliable data storage nodes but are as a group used to store data files over long periods of time with high reliably[51][52][53]. Figure 3.1 shows simple form of DSS.

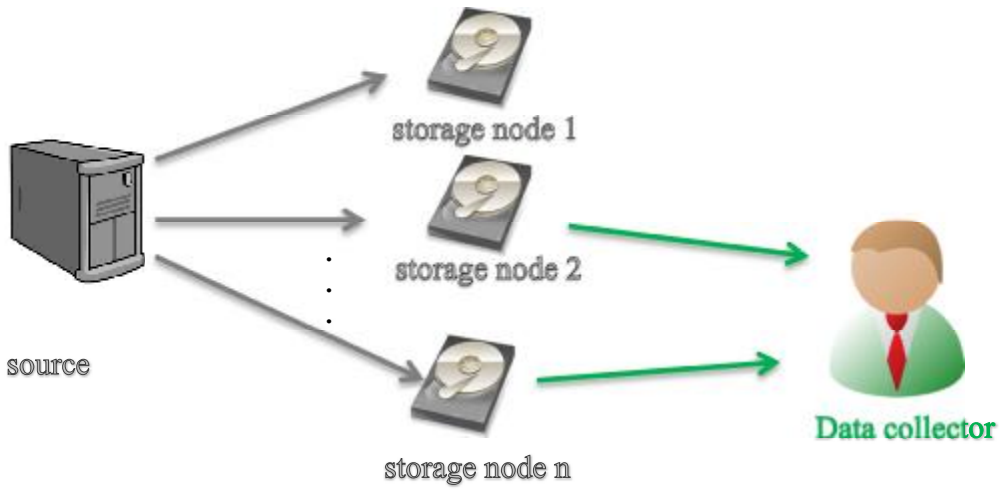


Figure 3.1: Distributed Storage Systems

In the case of fail node, a two solution can be used: either redundancy or repair, the system is repaired by replacing the failed node with a new “blank”

node. The new node connects to a remaining node which stores a same data at the failed node. So the receiver as yet can obtain the original information from any storage node [52] [34].

The reliability of this system came from using it redundancy especially when node fails. The redundancy has two schemes: first *replication* code second *erasure code* or a mix between them [34] [48] [54]. The details for these network codes are explained below.

a) Replication Code

This code is considered the simplest form of redundancy, which is used in many storage systems. The notion of replication code is to replicate data in many storage nodes in order to increase reliability of storage system. This scheme is low complexity from other codes because it isn't used any form of encryption of data before distributing it to each other storage node. For recover the file it only needs to combine the data from k nodes. Simple example of this code is suppose file of size M bits is spilt to k fragments at source node. Then distribute these fragments to many storage nodes [34] [55].

b) Erasure Code

The comparison between Erasure codes with the previous code is shown that the Erasure code is better in hand of storage and it can give high reliability with same amount of redundancy that is used in replication code but it requires high complexity in the implementation. The difference between these codes appears in using encoding on fragmented data. For example if a file is wanted to store it, the file is split into k fragments, the k fragments are encoded and then distributed into n storage nodes, when the file need to be constructed the k fragments must be recollected again [34] [48] [55].

3.4. Distributed storage system architectures

The main classifications of distributed storage system architectures are: *Client-Server* and *Peer-to-Peer* as shown in Figure 3.3. The architecture of the former is clear where entity may be behaves either client or a server, but cannot be both, while the participants can be both a client and a server at the same time in the second architectures a Peer-to-Peer [56]. The two schemes are explained below.

3.4.1. Client-Server based architecture

This scheme consists of two powerful computer *client*, and server, and these machines are connected through a network. The server is a computer responsible for providing authentication, consistency, backup, and servicing requesting clients. Clients are defined as simple machines. The main function of the server is to provide a service while the client consumes it. The communication between Client and server is done as follows, the client first begins the process by sending a request to the server and then waits for a reply message, when the server is received the request it will do the requested work and sends back a reply message to the client. This architecture has been widely used by distributed storage systems. The client-server scheme is shown in Figure 3.2.

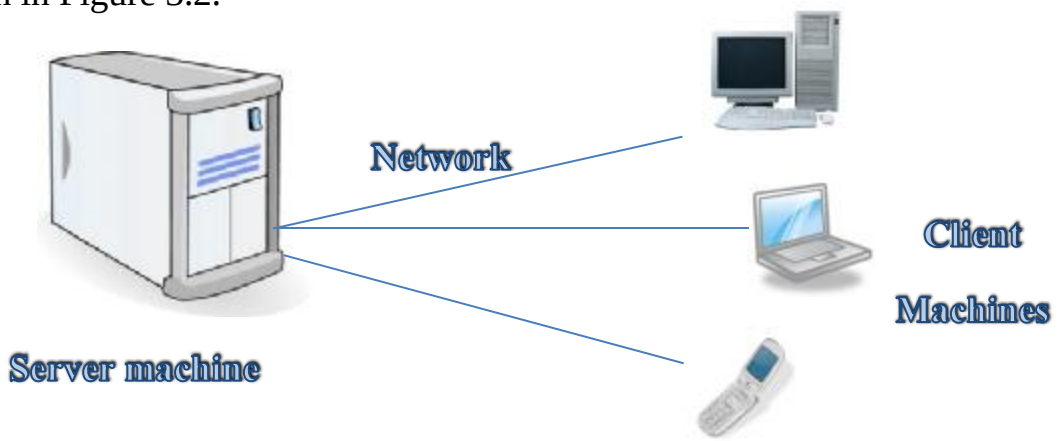


Figure 3.2: Client-Server System

If a network has number of clients exceeds the number of servers, the processing load for clients is reduced but the load of the server is increased.

The common examples of *client-server* paradigm are e-Mail services, web services, or domain name services. A disadvantage of this paradigm is the lack of scalability in case of large numbers of clients.

There is no central point control on functions of client-server architecture, but it has some level of centralization that is categorized to ***Globally Centralized*** and ***Locally Centralized***. In globally centralized architecture, the server is considered a central entity. This type is given limit in the scalability and made the architecture easy to failure; therefore a locally centralized architecture is proposed that is distributed responsibilities for many servers.

Client - server architecture is used in a controlled, trusted, and partially trusted environment. In order to achieve scalability and avoid the problem associated with operating in an ad-hoc untrusted environment such as the Internet, the Peer-to-Peer (P2P) methods are the better choices [56] [50].

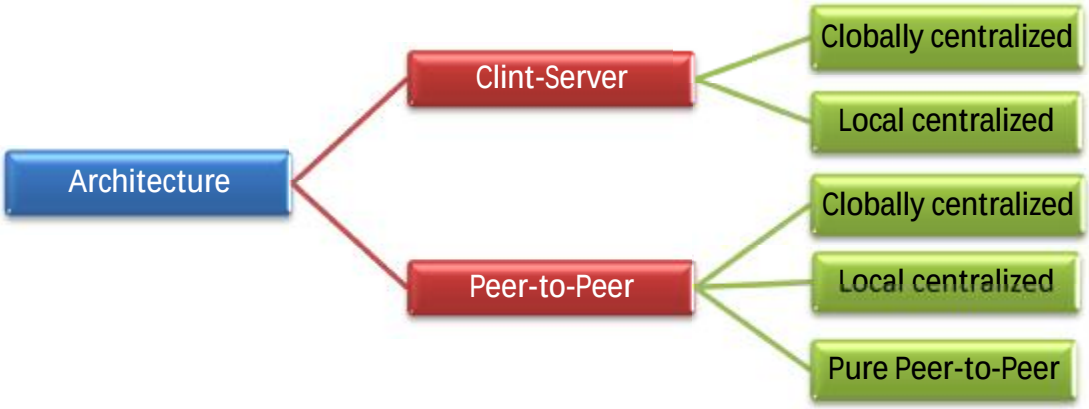


Figure 3.3: Architecture Taxonomy

3.4.2. Peer-to-Peer architecture

The term *peer-to-peer (P2P)* system is defined as many nodes (peers) connected to each other and combine their resources in order to achieve specific task. The peer can be server and a client simultaneously and can joins and leaves the network as they wish. The changes in the environment are not effecting on the network because a routing continually adapted it. In this scheme, the information is distributed among nodes instead of residing on a

single server as shown in Figure 3.4. The communication between peers differs than from client-server paradigm, where nodes make requests to other nodes and answer incoming calls. In the case of failing nodes, P2P systems still able to provide resources and functionality, and other nodes can tolerance for this failure because increase availability and robustness against failure in P2P systems.

This system gives higher scalability than previous systems. The most examples of applications that are used Peer-to-peer system are file-sharing programs (e.g. BitTorrent) and real-time communication (e.g. Voice or text chat).

P2P also contain or not a degree of centralization, it is categorized as ***Globally centralized, Locally centralized and Pure Peer-to-Peer.*** The globally centralization consists of central server contact all peers in the network, the details of these peers and their respective files contained in that central server. The weakness of global scheme is lacking scalability and susceptible of Single Point of Failure (SPF) problem.

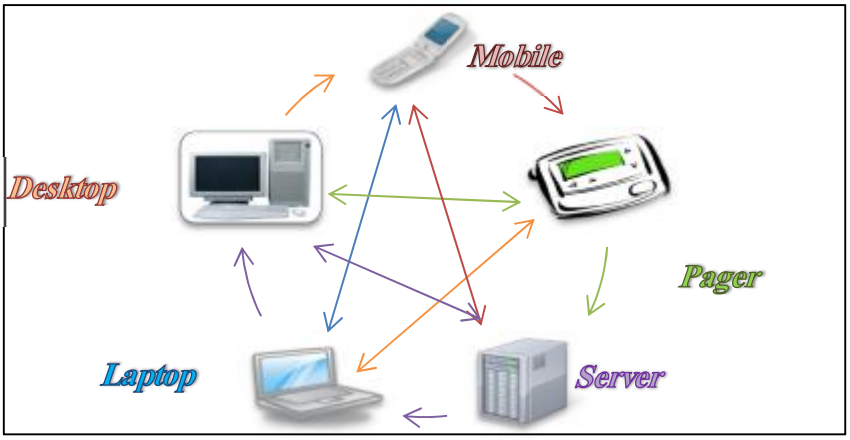


Figure 3.4: Peer-to-Peer Architecture

Locally centralized architecture is inspired by the shortcomings of early Peer-to- Peer efforts. To overcome the scalability bottleneck, locally centralized is used, this architecture is chosen few hosts that enjoy high performance and reliable characteristics these hosts are behave as *super*

nodes. These super nodes preserve a repository of meta-data (data about data) which a community of local nodes may query and update. Super nodes are communicated amongst each other forming bridges between communities, allowing local nodes to submit queries to a super node rather than broadcasting to the entire community. Super nodes form shape of centralization but the benefit of it was avoiding points of failure.

A pure Peer-to-Peer architecture distinguished with eliminate of any centralization, it is exhibited symmetrical harmony between all entities, this symmetrical nature made the system more scalable and efficient to adapt in a dynamic environment. Despite of the attractive features of this scheme but it also has some challenges such as the difficulty of achieving a symmetrical relationship between nodes especially in the asymmetric network such as the Internet.

Finally the goals were to build a system without centralization, establishing trust and accountability and work efficiently in an ad - hoc network, but these goals are difficult to be achieved without using some level of centralization neighborhood knowledge [50] [56].

3.5. Security of distributed storage

As previously mentioned, Data that has been dealt with is sensitive data that is a rapidly increased not only in the healthcare society but also in other areas such as customer records or financial data. It has become important to protect such data either while transmission or when it's stored [57].

The security and privacy of patient-related data are two essential components for the system security of the WBAN. The meaning of data security is that the data is securely stored and transferred while data privacy means the data can only be accessed by the people who have authorization to show and use it [58]. The security threats may lead a patient to a dangerous condition, and may to a death in case of medical applications.

To prevent malicious interaction with a WBAN, a strict and scalable security mechanism is required. The security and system performance in a WBAN are equally important, and thus, designing a low-power and a secure WBAN system is a primary challenge to the designers, therefor conjunction of a high-level security mechanism in a low-power and resource constrained sensor increases the computation, communication and management costs [21].

The main aspects of information security are *security attacks, security mechanisms, and security services* [59]. Any action that is compromised information security called "Security attack". While the Process that designed to detect, prevent or recover from security attack named "Security mechanism". If one or more security mechanisms are used to enhance system's security, this called "Security service" [60].

The important security issues related to storage are discussed. A comprehensive survey of the security services provided by the existing storage systems are presented, review of the existing solutions are explained and compared them. This is done in the remaining of this section and the next section.

3.5.1. WBAN Security Threats

Because of WBAN operates in the open access environments where various people can access to it [58], and the nature of wireless channel is vulnerable therefor different security threats that disturb the development of WBANs[61]. Healthcare network faced many different types of attacks or threats, these attacks can be categorized depending on the intended target either patient nodes or a health care system. The type of attack on the single server is attack by which malicious acts is specified on the node such as eavesdropping, modification, masquerading etc., while the attack on the distributed system is that the enemy aims to damage the connection between servers, such as Denial-of-Service (DoS), system intrusion, and impostor [59].

The description of these threats in WBAN as follows:

1) ***Eavesdropping***: The listening to communication channels called" eavesdropping" [62]. Attacker is eavesdropped on the packets from node to node, and can obtain the sensitive and valuable information by analyzing the packets [61].
2) ***Data Modification***: The eavesdropped information can be removed or replaced partially or fully by attacker, and later, the attacker sends the modified information back to the original receiver in order to achieve some illegal purpose [61]. This type of attack effect on the safety. For example, if an attacker modifies health information, in this case a wrong disease is gotten. This is considered a great danger [63].
3) ***Denial of Service***: When network traffic has exceeded the capacity of the systems, DoS attack will occur. This attack related to the effects of both intentional acts of malicious or compromised nodes and unintentional exaggerated peak network usage [61]. Denial of service attack makes the resource of healthcare computer unavailable to its intended users [62].
4) ***Masquerading***: Is that an entity pretending to be a different entity in order to obtain unauthorized access of health care data [62].
5) ***Jamming\tampering***: The interference with the radio frequencies of the nodes named Jamming. The physical tampering in WBAN is less due to the nodes that are always in very near to the human body. As a result of tampering attacker, the patient information is acquired by damage, replace, and electronically interrogate the WBAN nodes [21].
6) ***Spoofing*:** This is believed as the most common routing attacks. An adversary can complicate the network by creating routing loops, eclectic forwarding and also adversary can stop packet forwarding when includes a node in a data flow path [21].

3.5.2. Security Services

The proposed system tries to ensure the following security services:

1) ***Authentication and Authorization:*** when any system is built, it should support these services. The authentication is defined as "entity authentication" if it is verifying the identity of an entity, or called "message authentication" if verify the source of a message. The authentication is done in DSS when the servers begin to communicate. The goal of Authentication is to prevent a malicious user from masquerading as a trusted network node. While the authorization means the act of giving privileges to the users. Authentication can be performed using various techniques such as passwords, digital signatures or Message Authentication Codes (MAC). Authorization can be performed by maintaining Access Control List (ACL) or by using capability certificates that list the access rights bestowed to the holder of the certificate [57] [59].
2) ***Availability***: The task of availability service ensures that the patient information is always available to the physician [21]. Availability service found in order to remind patient-related accessible even if the server failures. Availability is performed by using replication or redundancy [57] [58].
3) ***Confidentiality***: It protects data from an illegal user. The methods that are used to provide data confidentiality are: encryption and ACL [63]. The amount of confidentiality is determined depending on the type of cipher scheme and mode of operation [64].
4) ***Data Integrity*:** Data Integrity service guarantees the correctness of data, protecting the data against modification, deletion, creation and replication from an unauthorized user. The methods that provide data integrity are: digital signatures and MAC [63]. The message is

discarded by any receiver if a malicious entity modifies exchange message [65].

5) ***Data Freshness*:** This technique introduces support to the data confidentiality and integrity where without data freshness these services are not enough. The function of data freshness is ensures that the data is fresh [21]. This security service prevents the attacker from replaying the old frames that he/she eavesdropped [61].
6) ***Non-repudiation*:** By using this service the source that generated patient-related data cannot be disowned it [58].
7) ***Tolerance to message loss*:** The receiver must be aware of the message loss with respect to the noise and the existence of the attacker in the communication channel. The number of the lost messages can be detected by using a counter [64].

3.6. WBAN Security Solutions

After the data is produced, transferred, and stored at one or more remote storage servers; it becomes susceptible to many kinds of attack whether unauthorized disclosures, unauthorized modifications or replay [57]. Thus it is important to use strong security mechanisms that are meeting the required service.

Few security solutions that are proposed for the wireless sensor network can be used to provide a security solution for the WBAN, such as Security Protocols for Sensor Networks (SPINS) that are provided data confidentiality, authentication and freshness [38].

Some security solutions that are used in the proposed system are achieving important security services. These solutions are explained in the following sections.

3.6.1. Residue Number System and Redundant Residue Number System

The basic mathematical steps that are used for converting any weighted number system to residue number system (RNS) and Redundant Residue Number System (RRNS) are explained within this section [66] [67].

A RNS depends on choosing *h* pairwise, positive and relative primes $m_1 \dots m_h$ that is called moduli or prims, and the range of these moduli calculated by the equation below:

$$\boldsymbol{M = \prod_{p=1}^{h} m_p} \quad (1)$$

These moduli did not lose generality where $m_p > m_{p-1}$, $p\epsilon[2,h]$. Any non-negative integer *X* can be represented as follows:

$$\boldsymbol{x_p = X \bmod m_p} \quad (2)$$

x_p is the residue of *X* and $x_p < m_p$, thus any integer represented from $X \xrightarrow{RNS} (x_1, \dots, x_p)$, $p\epsilon[1,h]$ residues. The legitimate range of *X* is $[0, M)$.

The RNS is designed with some number of redundant moduli in order to control the error, whether ***error-correction*** and ***error-detection***. It becomes called ***RRNS*** that is widely employed in the space of error control. It is worked as follows, choosing $(h+r)$ moduli $m_1, m_2, \dots\dots, m_h, m_{h+1}, .., m_{h+r}$. The moduli $m_1, m_2, \dots\dots, m_h$, are termed as non-redundant moduli, while the additional *r* moduli $m_{h+1}, .., m_{h+r}$, are the redundant moduli where the range of the system is still calculated by applying Eq. (1). While the redundant rage is calculated by the equation below:

$$\boldsymbol{MR = \prod_{p=h+1}^{r} m_p} \quad (3)$$

These moduli don't loss generality where $m_p > m_{p-1}$. $p\epsilon[2,h]$. For non-negative integer *X*, the residues of *X* can be found by applying Eq. (2).

The (h + r)-tuples$(x_1, \dots, x_{h+r})$, are called residue representation of X, where $X \xrightarrow{RRNS} (x_1, \dots, x_{h+r})$. The residue digits $x_1, \dots, x_h$, are the non-

redundant residue digits, while $x_{h+1}, \ldots, x_{h+r}$, and are the redundant residue digits. RRNS could provide representations to all integers in the range [0, M·MR), the dynamic range of RRNS is separated into two domains: legible and illegible, the legible range of representation is limited to [0, M), and the corresponding $(h + r)$ –tuples are called legible while the integers in [M, M·MR) range and the corresponding $(h + r)$–tuples are called illegible.

RRNS achieve dependable and secure data storage in wireless networks, where is encode the data into $(h+r)$-tuple of residues using $h+r$ moduli. These residues are distributed among the mobiles in the network. Recovering the original information requires the knowledge of at least h residues and of the corresponding moduli. Data can be reconstructed in the presence of up to $s \leq r$ residue losses (erasures), combined with up to $\frac{r-s}{2}$ corrupted residues[68].

Example: Assume that the moduli set m_1=5, m_2=7, and m_3=9 are the non-redundant, and the redundant moduli m_4=11, m_5=13, m_6=16. The dynamic range of these moduli according to Eq. (1) is [0, 314], where $M = 5 * 7 * 9$. While the illegitimate range is given by [315, 720720] since $MR = 11 \times 13 \times 16 = 2228$, Table 3.1 illustrates the RNS and RRNS representation of integer messages X and the corresponding residues that are calculated by applying Eq. (2). Such as X=17, the residues representation of this integer are: $x_1 = 17 \bmod 5 = 2$, $x_2 = 17 \bmod 7 = 3$, …, $x_6 = 17 \bmod 16 = 1$.

Table 3.1: RNS and RRNS representation of integer messages

Message	**RNS**			**RRNS**		
X	x_1	x_2	x_3	x_4	x_5	x_6
X_1 = **2**	2	2	2	2	2	2
X_2 = **5**	0	5	5	5	5	5
X_3 = **17**	2	3	8	6	4	1
X_5 = **274**	4	1	4	10	1	2
X_6 = **320**	0	5	5	1	8	0

It can be seen that both $X = 5$ and $X = 320$ have the same Non-redundant Residue representation but different Redundant Residue. This is because the integer X = 320 is in the illegitimate range and thus has no unique representation by the information moduli, i.e. m_1, m_2, m_3.

The basic of the RNS and RRNS techniques is choosing the moduli (primes) the characteristics of these primes are: pairwise, positive, relative primes, and did not lose generality. [68] Built library of moduli meets all these requirements as shown in the Table 3.2.

Table 3.2: - Moduli Library

m1	65536	**m10**	65503	**m19**	65449	**m28**	65393	**m37**	65339
m2	65533	**m11**	65501	**m20**	65447	**m29**	65383	**m38**	65327
m3	65531	**m12**	65497	**m21**	65437	**m30**	65381	**m39**	65323
m4	65529	**m13**	65491	**m22**	65431	**m31**	65371	**m40**	65321
m5	65527	**m14**	65489	**m23**	65423	**m32**	65369	**m41**	65311
m6	65525	**m15**	65479	**m24**	65419	**m33**	65363	**m42**	65309
m7	65521	**m16**	65477	**m25**	65413	**m34**	65357	**m43**	65293
m8	65519	**m17**	65473	**m26**	65411	**m35**	65353	**m44**	65287
m9	65509	**m18**	65459	**m27**	65407	**m36**	65347	**m45**	65281

Table 3.3 is shown another library of moduli. The size of primes in this library is large compare to the previous library.

3.6.2. Code Efficiency of dependable and secure data storage

The code efficiency (CE) of the dependable and secure data storage is depended on the moduli that are used in the encoding process and depend on the using redundancy or not. The large prim in the Table 3.2 is $65536=2^{16}$ this mean the data is divided every 16 bit. While the large moduli in the Table 3.3 is $6461333947 \cong 2^{32}$, mean the data is divided into every 32 bits. CE is calculated according to the Eq. (4). [68].

$$\emptyset = \boldsymbol{b/e} \quad (4)$$

Where $b = \log M$, and $e = 16(h + r)$ or $e = 32(h + r)$

Table 3.3: - Moduli Library with big prims

m1	6461333093	**m16**	6461333419	**m31**	6461333759
m2	6461333101	**m17**	6461333423	**m32**	6461333773
m3	6461333117	**m18**	6461333491	**m33**	6461333783
m4	6461333143	**m19**	6461333521	**m34**	6461333797
m5	6461333171	**m20**	6461333537	**m35**	6461333831
m6	6461333227	**m21**	6461333563	**m36**	6461333839
m7	6461333267	**m22**	6461333621	**m37**	6461333849
m8	6461333269	**m23**	6461333647	**m38**	6461333849
m9	6461333311	**m24**	6461333651	**m39**	6461333863
m10	6461333321	**m25**	6461333653	**m40**	6461333867
m11	6461333353	**m26**	6461333687	**m41**	6461333887
m12	6461333357	**m27**	6461333693	**m42**	6461333899
m13	6461333369	**m28**	6461333713	**m43**	6461333917
m14	6461333387	**m29**	6461333737	**m44**	6461333929
m15	6461333399	**m30**	6461333741	**m45**	6461333947

3.6.3. Decoding Redundant Residue Number System

In order to recover non-negative integer *X* from its residues, inverse processes are needed. Conventional techniques are used. The CRT is utilized for inversion transform, as well as alternative techniques like BEX with MRC, and new CRT I are used to same purpose.

A. Chinese Remainder Theorem (CRT)

The Chinese Remainder Theorem was the first method that works on modular arithmetic proposed at the fifth century by Sun Tzu. But the use of this arithmetic to represent numbers was introduced only in 1959 by H.L. Garner.2. CRT is used in RRNS in order to reconstruct any integer from it residues by apply the equation below [69]:

$$\boldsymbol{X} = \left(\sum_{p=1,h} \boldsymbol{x_p} \frac{\boldsymbol{M}}{\boldsymbol{m_p}} \boldsymbol{b_p}\right) \boldsymbol{mod\ M} \qquad (5)$$

For each p∈ [1, h], b_p, is multiplicative inverse of M/mp modulo mp, that is $\left(b_p \frac{M}{mp} \right) mod\ m_p = 1$.

Euclid's Gcd algorithm used to find multiplicative inverse (MI). The definition of the function MI assumes that the first argument is the number whose MI you want to calculate and the second argument the modulus. The MI exists only when gcd(num,mod) = 1. M/m_p is num while the m_p is mod[70].

B. Base extension (BEX) with mixed radix conversion (MRC)

In spite of CRT is a classical algorithm but the implementation of it is computationally intensive for large moduli values as deals with modular operations with a large value of range M [67]. In order to avoid processing of large integers, the alternative method that is widely used is the base extension (BEX) operation in conjunction with the Mixed Radix Conversion (MRC) method [71]. MRC is formulated as follows:

Given a set of residues (x_1, x_2, x_3, . . ., x_h) defined on the corresponding set of moduli (m_1, m_2, m_3, . . ., m_h) and a set of mixed radix digits (a_1, a_2, . . ., a_h), the decimal equivalent of the residues can be determined as follows in Eq. (6) to Eq. (7):

$$\boldsymbol{X = a1 + a2m1 + a3m1m2 + \cdots + ah \prod_{i=1}^{h-1} mi} \quad (6)$$

Where the Mixed Radix Digits (MRD) are given as:

$$\boldsymbol{a_1 = x_1}$$

$$\boldsymbol{a_2 = \left((x_1 - a_1){m_1}^{-1}\right) m_2}$$

$$\boldsymbol{a_h = (\left(\left(\ldots\left((x_h - a_1){m_1}^{-1} - a_2\right){m_2}^{-1} - \cdots -\right) a_{h-1}\right) {m_{(h-1)}}^{-1}) m_h} \quad (7)$$

The mixed radix digits limited between the periods $0 \leq a_i < m_i$.

C. New Chinese Remainder Theorem (CRT I)

New Chinese Remainder Theorems (CRT I) is a modified version of the traditional CRT, it is designed to make the computations faster and efficient without any overheads where the CRT demand a slow large moduli operation while the MRC requires finding the MRD which is a slow process. This is based on the idea "the size of the numbers is directly proportional to the delay of the operations", and therefore smaller numbers imply faster operations. In the CRT I, the weighted number can be retrieved faster because the operations are done in parallel, without depending on other results. The conversion of CRT I overcomes the bottleneck of using traditional CRT and MRC techniques. This Theorem operates as follow:

Given the residue numbers (x_1, x_2, x_3, x_h), with corresponding moduli (m_1, m_2, m_3, m_h) the weighted number X can be computed using the following equation[72].

$$\boldsymbol{X} =$$

$$[\boldsymbol{x_1} + \boldsymbol{k_1 m_1}(\boldsymbol{x_2} - \boldsymbol{x_1}) + \boldsymbol{k_2}\, \boldsymbol{m_1}\, \boldsymbol{m_2}(\boldsymbol{x_3} - \boldsymbol{x_2}) + \cdots +$$

$$\boldsymbol{k_{h-1}}\, \boldsymbol{m_1}\, \boldsymbol{m_2} \ldots \boldsymbol{m_{n-1}}\, (\boldsymbol{x_h} - \boldsymbol{x_{h-1}})]\boldsymbol{mod}\, \boldsymbol{m_1 m_2} \ldots \boldsymbol{m_{h-1} m_h} \quad (8)$$

Where $k_1 = (m_1)^{-1} mod\ m_2 m_3 \ldots m_h$,

$k_2 = (m_1 m_2)^{-1} mod\ m_3 m_4 \ldots m_h$, and similarly

$$\boldsymbol{k_{(h-1)}} = (\boldsymbol{m_1 m_2} .. \boldsymbol{m_{h-1}})^{-1} \boldsymbol{mod}\, \boldsymbol{m_h} \quad (9)$$

3.6.4. Rivest-Shamir-Adleman (RSA) algorithm

This is one of algorithms that is proposed for public-key cryptography introduced by "Rivest, Shamir, and Adleman". The procedures of this algorithm for encryption and decryption messages are [73]:

a. Suppose *p* and *q* are two (large) distinct primes $p \neq q$

b. Calculate $n = p \times q$

c. Calculate $\emptyset(\mathrm{n}) = (\mathrm{p}-1)(\mathrm{q}-1)$

d. Select *integer e* $\mathrm{Gcd}(\emptyset(n), e) = 1; 1 < e < \emptyset(n)$

e. Determine privet key *d* such that $d \equiv e^{-1}(mod\ \emptyset(n))$ (10)

Public key $PU = \{e, n\}$

Private Key $PR = \{d, n\}$

The Encryption process includes taking plain text M < n, and fined the cipher text by applying below equation:

$$C = M^e \ mod\ n \quad (11)$$

While the Decryption process on cipher to recover plaintext as follow equation:

$$M = C^d \ mod\ n \quad (12)$$

3.6.5. Hash functions

To ensure that the message is not modified during transmission or storage, some algorithms can be used such as DSA (Digital Signature algorithm) this algorithm uses a hash function (SHA-1) mixed with its steps. Hash function used alone to provide integrity.

The SHA-1 algorithm is explained firstly and the DSA is shown at the end. SHA-1 hash function *H* may be used to hash a message *M*, having a length of *L* bits, where$0 \leq L < 2^{64}$ and transforms it to produce a hash value *h* that is a function of the message $h = H(M)$, the input is a variable string and the result of SHA-1 is a 160-bit message digest.

The hash value is also called a message digest or a fingerprint of the message because there is a very low probability that two messages will output the same hash value. Hash functions are hard to invert. Overview on the processes of SHA-1 algorithm is explained below [74] [75]:

1) A message schedule of eighty 32-bit words labeled *W*0, …, *W*79

2) Five working variables of 32 bits labeled ***a***, ***b***, ***c***, ***d***, and ***e***.

3) A hash value of five 32-bit words labeled $H_0^{(i)}, H_1^{(i)}, H_2^{(i)}, H_3^{(i)}, H_4^{(i)}$,which will hold the initial hash value,H^0 replaced by each successive intermediate hash value (after each message block is processed) H^i, and ending with the final hash value, $H^{(N)}$,SHA-1 also uses a single temporary word, *T*.

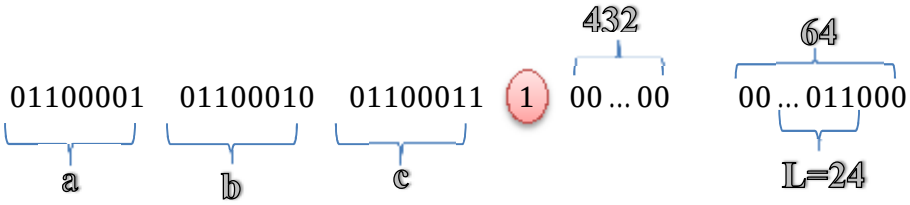
SHA-1 Preprocessing

1. Set the initial SHA-1 hash value, $H^{(0)}$, shall consist of the following five 32-bit words, in hex:

$$H_0^{(0)} = 67452301$$
$$H_1^{(0)} = efcdab89$$
$$H_2^{(0)} = 98badcfe$$
$$H_3^{(0)} = 10325476$$
$$H_4^{(0)} = c3d2e1f0$$

2. **Padding**: The purpose of this padding is to ensure that the padded message is a multiple of 512 or 1024 bits, depending on the algorithm. Suppose that the length of the message *M* is *L* bits. Append the bit "1" to the end of the message, followed by *k* zero bits, where *k* is the smallest, non-negative solution to the equation $L + 1 + k \equiv 448 \bmod 512$. Then append the 64-bit block that is equal to the number L expressed using a binary representation. For example, the (8-bit ASCII) message "**abc**" has length 8×3 = 24 , so the

message is padded with a one bit, then 448− (24 +1) = 423 zero bits, and then the message length, to become the 512-bit padded message



SHA-1 Hash Computation

The SHA-1 hash computation uses the fallowing functions and constants.

SHA-1 Functions

SHA-1 uses a sequence of logical functions $f0, f1, \dots, f79$. Each function ft,where $0 \leq t < 79$, operates on three 32-bit words, *x*, *y*, and *z*, and produces a 32-bit word as output. The function $ft\ (x, y, z)$ is defined as follows:

$$ft\ (x,y,z) = \begin{cases} Ch(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z) & 0 \leq t \leq 19 \\ Parity(x,y,z) = x \oplus y \oplus z & 20 \leq t \leq 39 \\ Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) & 40 \leq t \leq 59 \\ Parity(x,y,z) = x \oplus y \oplus z & 60 \leq t \leq 79. \end{cases}$$

SHA-1 Constants

SHA-1 uses a sequence of eighty constant 32-bit words, $K0, K1, \dots, K79$, which are:

$$Kt = \begin{cases} 5a827999 & 0 \leq t \leq 19 \\ 6ed9eba1 & 20 \leq t \leq 39 \\ 8f1bbcdc & 40 \leq t \leq 59 \\ ca62c1d6 & 60 \leq t \leq 79 \end{cases}$$

Each message block $M^{(1)}, M^{(2)}, \dots, M^{(N)}$is processed in order, using the following steps:

For $i = 1$ to N:

{

1. Prepare the message schedule,{W}:

$$W_t = \begin{cases} M_t^{(i)} & 0 \leq t \leq 15 \\ ROTL^1 = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) & 16 \leq t \leq 79 \end{cases}$$

2. Initialize the five working variables, *a*, *b*, *c*, *d*, and *e*, with the $(i-1)^{\text{st}}$ hash value

$$\begin{aligned} \mathrm{a} &= \mathrm{H}_0^{(\mathrm{i}-1)} \\ \mathrm{b} &= \mathrm{H}_1^{(\mathrm{i}-1)} \\ \mathrm{c} &= \mathrm{H}_2^{(\mathrm{i}-1)} \\ \mathrm{d} &= \mathrm{H}_3^{(\mathrm{i}-1)} \\ \mathrm{e} &= \mathrm{H}_4^{(\mathrm{i}-1)} \end{aligned}$$

3. *for t=0 to 79:*

{

$$\begin{aligned} T &= ROTL^5(a) + f_t(b,c,d) + e + K_t + W_t \\ e &= d \\ d &= c \\ c &= ROTL^{30}(b) \\ b &= a \\ a &= T \end{aligned}$$

}

4. *Compute the* i^{th} *intermediate hash value* $H^{(i)}$

$$\begin{aligned} H_0^{(i)} &= \mathrm{a} + \mathrm{H}_0^{(\mathrm{i}-1)} \\ H_1^{(i)} &= \mathrm{b} + \mathrm{H}_0^{(\mathrm{i}-1)} \\ H_2^{(i)} &= \mathrm{c} + \mathrm{H}_0^{(\mathrm{i}-1)} \\ H_3^{(i)} &= \mathrm{d} + \mathrm{H}_0^{(\mathrm{i}-1)} \\ H_4^{(i)} &= \mathrm{e} + \mathrm{H}_0^{(\mathrm{i}-1)} \end{aligned}$$

}

After repeating steps of *N* times (i.e., after processing $M^{(N)}$), the resulting 160-bit message digest of the message, *M*, is*:*

$$M_0^{(N)} || M_1^{(N)} || M_2^{(N)} || M_3^{(N)} || M_4^{(N)}$$

3.6.6. Digital Signature Algorithm

The DSA algorithm is summarized in following points [60]:

The Global Public-Key Components are

- p prime number where $2^{L-1} < p < 2^{L}$ for $512 \leq L \leq 1024$
- q prime divisor of (p _ 1), where $2^{159} < q < 2^{160}$
- $g = h^{(p-1)/q} mod\ p,$ (13)

where h is any integer with $1 < h < (p-1)$

User's Private Key

- x random or pseudorandom integer with $0 < x < q$

User's Public Key

- $y = g^{x} \bmod p$ (14)

User's Per-Message Secret Number

- k= random or pseudorandom integer with $0 < k < q$

Signing

- $r = (g^{k} mod\ p) mod\ q$ (15)
- $s = [k^{-1}(H(M) + xr)] mod\ q$ (16)
- Signature = (r, s)

Verifying

- $w = (s')^{-1} mod\ q$ (17)
- $u_1 = [H(M')w] mod\ q$ (18)
- $u_2 = (r')w\ mod\ q$ (19)
- $v = [(g^{u_1} y^{u_2}) mod\ p] mod\ q$ (20)

Test: $v = r'$

M = message to be signed

H (M) = hash of M using SHA-1

M', r', s' = received versions of M, r, s

3.7. Public-key cryptography performance

The use of public-key cryptosystems can be classified into three categories

- Encryption /decryption.
- Digital signature.
- Key exchange.

Table 3.4 indicates the difference between most common public key algorithms that are suitable for achieving all or some of these categories.

Table 3.4: Applications for Public-Key Cryptosystems

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No
ElGamal	Yes	Yes	No

The RSA algorithm and digital signatures are used to provide authentication, non-repudiation (A sender should not deny having sent and signed a message), and Integrity. It is important to emphasize that there are no actual energy limitations that can prevent using the RSA (or other public key algorithms) in the distributed system if used for WBAN. Despite their relatively high computing energy requirements, such public key algorithms are only used in the upper layer of the network hierarchy that includes various system servers. In other words, the RSA is not applied on the lower level of the network that incorporates sensors with critical energy constraints. The security techniques for achieving integrity and authentication at sensor level are out of scope of this work.

Chapter Four

Proposed System design And Implementation

Chapter Four

Proposed System design and Implementation

4.1. Introduction

This chapter gives details about algorithms of the proposed distributed storage system used to store the patient information at the top level of WBANs. The proposed system is explained here in details. A collection of programs are utilized in system simulation. These programs are executed in Primary Server, Storage Server and finally in Reader Server. This implies that a network connection is used in this system. Due to the lack of possibility for creating large network, the simulation has been performed in a prototype local area network (LAN).

Microsoft Visual C# has been used in order to write the software because it enjoys many characteristics such as easy for implementation of graphical user interface, and it is considered one of the top of the currently used programming languages.

4.2. System Initialization

Before beginning to explain the specifics of the system, the initialization process must be illustrated. Initialization consists of two phases:

- ***Database Building Phase:*** In this phase simple Microsoft Access Database 2010 is built.
- ***Creation of Network Connection Phase:*** This phase responsible for creating local area network.

The details of these phases are explained in the next sections.

4.2.1. Database Building Phase

This DB like somewhat these found in the hospital. It contains two tables. The former table is called the patient and the second is doctor table. The patient table is contained many records represent the patients, and fields represented (ID, PN (Patient Number), Name, Temperature, Age, Heart pulse, Room No. and Date), Table 4.1 shows the patient fields with example.

Table 4.1: Patient Table

ID	PN	Name	Temperature	Age	Heart pulse	Room No.	Date
1	100	Abeer Dawood Salman	37	23	72	122	01/06/2013

While the doctor table contains ID records with the following fields (drug name and doctor name... etc.) as shown in Table 4.2.

Table 4.2: Doctor Table

ID	Drug name	Doctor name
1	Epinephrine	Jamal Omer Ali

The primary key of two tables is ID field. This primary key is used to create Relationship between above tables; this means is that ID is same in two tables. The result of this relationship is collecting all fields that are contained in patient and doctor tables in one query as shown in in Table 4.3:

Table 4.3: Query

ID	PN	Name	Age	Temper-ature	Heart pulse	Room No.	Drug name	Doctor name	Date
1	100	Abeer Dawood Salman	23	37	72	122	Epinephrine	Jamal Omer Ali	01/06/2013

This DB is not constant, it is built just as model in contact with the software on the primary server, you can design another DB with different fields as you require and may there are differences between this DB and these found in the medical server.

4.2.2. Creation of Network Connection Phase

The second phase of initialization is creating the local connection between the servers. The process of creating network connection can be summarized in the steps that are shown in Figure 4.1.

Note that these steps are applied in Microsoft Windows 7 environments. After creating this network by primary server, the PS must connect to it and wait other servers to connect with it after entering the right secret key, this the first authentication process in the system before beginning exchange information process. To complete the communication process rightly the IP of each server is determined at the outset.

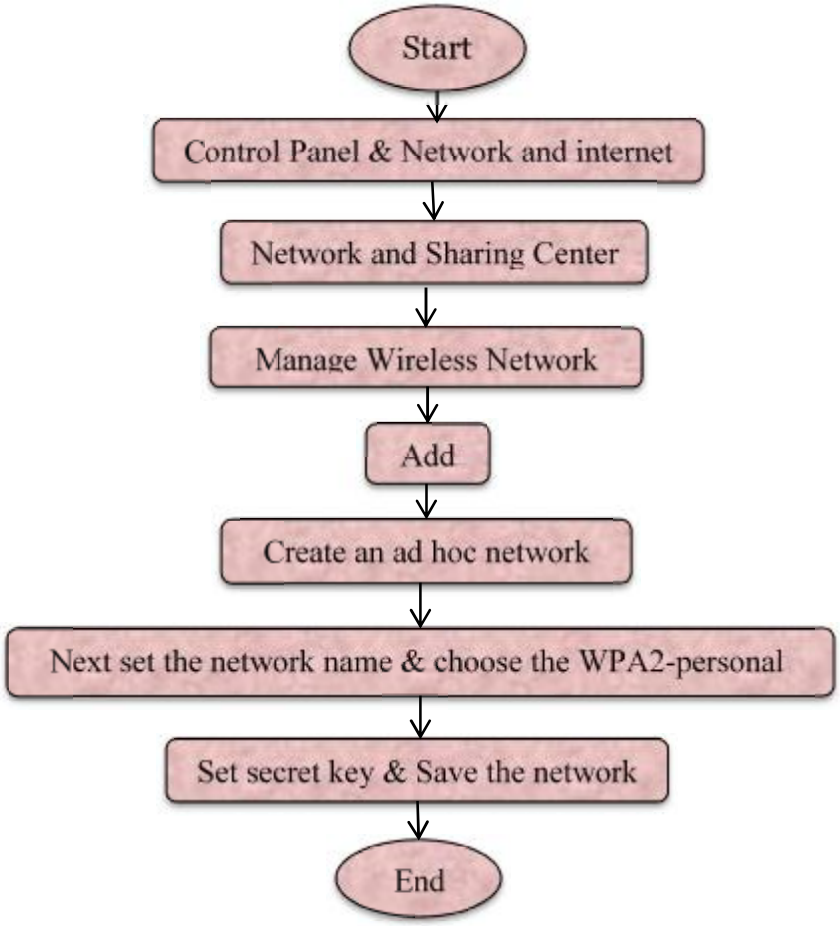


Figure 4.1: Creation of Network Connection Phase

4.3. System Model

The main phases of the proposed system are depicted in Figure 4.2. The overall system consists of:

- *One Primary Server* (*PS*).
- *Collection of Storage Servers (SS*).
- *One or more Reader Servers (RS)*.

PS is a main server reside in the hospital contains DB for patients. SS_s are server spread in the network used to store the patient data after encoding it by PS using RRNS. RS is server need to read the patient data therefore it must collect these data from SS_s and apply one of the decoding algorithms for this purpose; the system's phases are distributed among all of these servers. Each phase is taken alone and shown how it is designed and implemented in the simulation.

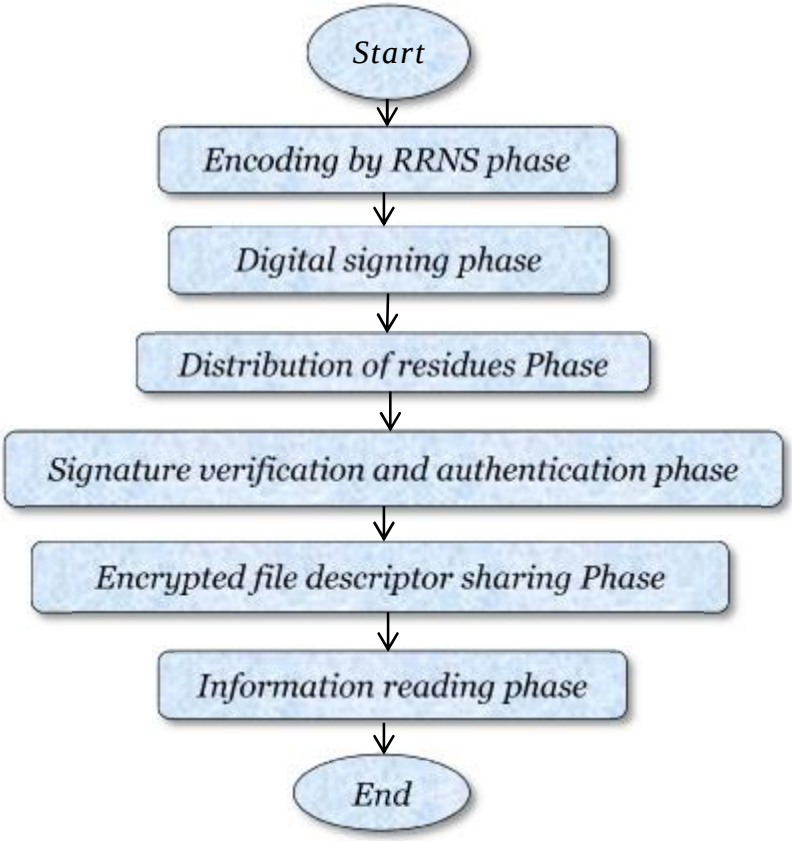


Figure 4.2: System Model Phases

The general overview of the proposed system abstracted in the Figure 4.3.

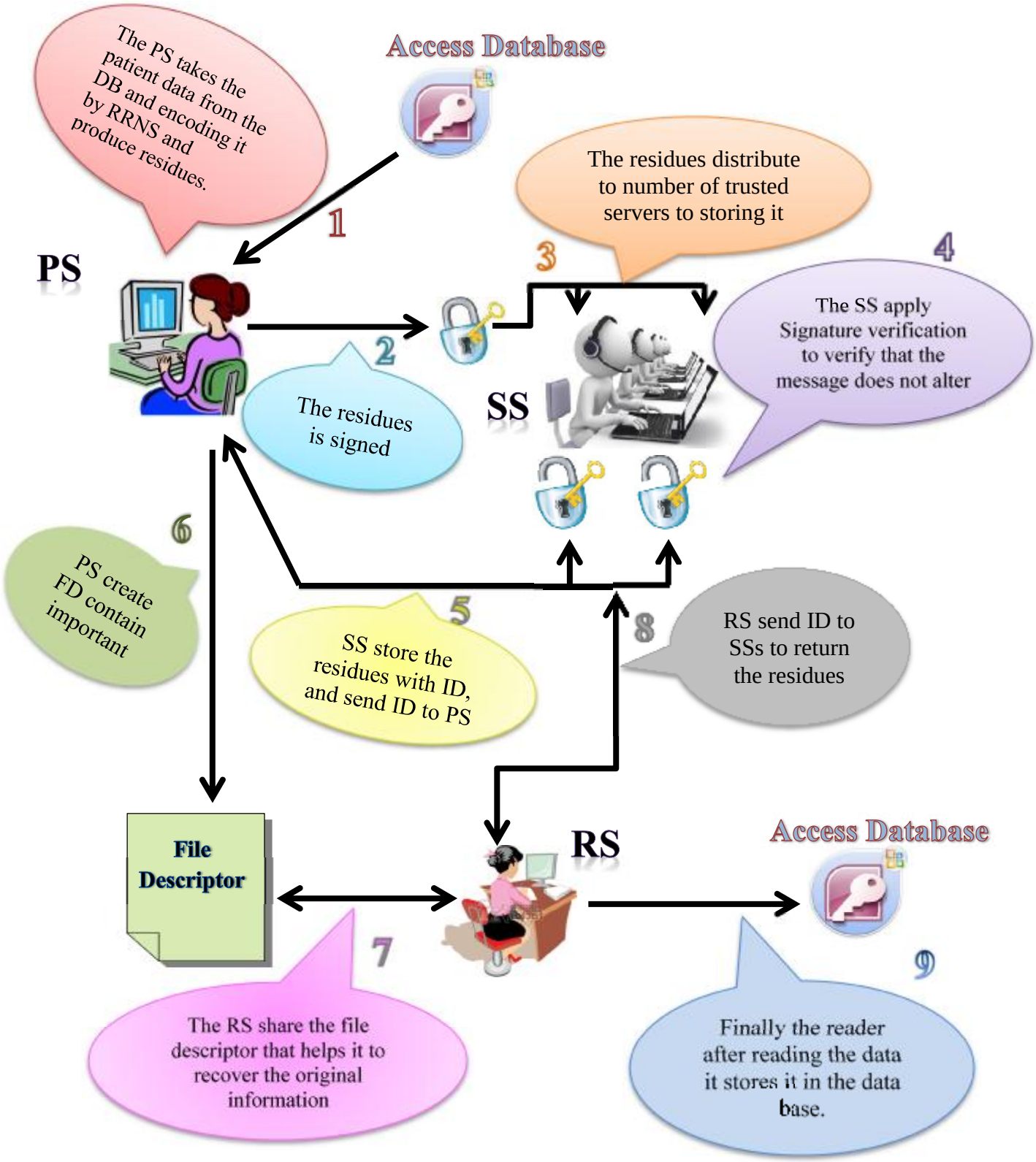


Figure 4.3: General Overview

4.3.1. Encoding By RRNS Phase

Inside of this phase, the built database is connected with the program by using the *Data.OleDb* class. The software is executed for each record in the DB that represents the patient data as follows:

1. The library of moduli is chosen. Two libraries can be used in the system: One is called ″Moduli Library" shown in Table 3.2 and the other library is called ″Moduli Library with big prims″ shown in Table 3.3. These moduli are used in the encoding and decoding process. Numbers of moduli are selected depending on these libraries.
2. Set the parameters (*h*, *r* and *bytes*) that are the parameters specified for RRNS where the *h* is non-redundant residue, *r* is redundant residue and *byte*s represent the number of the characters that are based on it when the information is split.
3. Take the patient data from DB that is represented by Query and converted it into binary form and put the result in *'arayofbinary'* with solving the problem of size that is accrued when number of character in Query %byte≠0.
4. Call the RRNS () function that accepts *'arayofbinary'* as arguments. This function works as follows:
    a. Find the number of the records according to Eq. (21) for each patient's data that are generated after splitting it.

$$\boldsymbol{S} = \frac{\boldsymbol{number\ of\ charecters\ in\ Query}}{\boldsymbol{bytes}} \qquad (21)$$

    b. Take every (bytes*8) bits from binary array and convert it into decimal form by calling *converttodecima ()* function that took this array, and return a decimal number stored in *'decimal'* array, as the below format:

```
decim: converttodecima (binary array)
```

    c. This function does some printing arrangements.

5. Calling the *module ()* function, this function is responsible for:
   a. Selecting (h+r) moduli from the library of prims.
   b. Finding the range of these moduli by applying the Eq. (1) mention in the chapter3.
   c. If the decimal numbers less than the moduli, the corresponding residue will be equal to decimal content and the storage system will be displayed. To solve this problem random number<range is added to these decimal digits.
6. Here the *residufun ()* function is summoned. This function responsible for finding the residues by applying the Eq. (2), the format of it is:

`residufun (decimal array)`

The result of apply above steps on the Query is explained in the Table 4.4 with *h*=2, *r*=1, *byte*=2 and the m_1=65281, m_2=65287, and m_3=65293. Query size is 66 character, therefore number of record *s*=33. Range $M=m_1.m_2.m_3$, M=4262000647, the random number is=1771908380.

Table 4.4: Encoding Process

Record	***Content***	***Binary form***	***Decimal content***	***Decimal content***	x_1	x_2	x_3
b_1	10	0011000100110000	12592	1771920972	64070	31792	64831
b_2	0a	0011000001100001	12385	1771920765	63863	31585	64624
b_3	be	0110001001100101	25189	1771933569	11386	44389	12135
b_4	er	0110010101110010	25970	1771934350	12167	45170	12916
b_5	d	0010000001100100	8292	1771916672	59770	27492	60531
b_6	aw	0110000101110111	24951	1771933331	11148	44151	11897
b_7	oo	0110111101101111	28527	1771936907	14724	47727	15473
. . **b_{37}**	. . 12	. . 0011000100110010	. . 12594	. . 1771920974	. . 64072	. . 31794	. . 64833

As a result of applying all these functions are: $s.(h+r)$ residues that will propagate to the SS_s later. The distribution process is done without constraining only different residues belong to same record distribute to a different server. After the first phase is completed. It can be summarized in the algorithm that is shown in Figure 4.4 below.

Algorithm Name: Encoding By RRNS
In put: patient data
Out put: residues

Step1: Start
Step2: Bulit DB contains patient data
Step3: choosing library of moduli
Step4: Set RRNS's parameters (*h*,*r*,*byte*,*modules*,*pateint data*)
Step5: Execute RRNS function as follow:

- Split data into *s* records with bytes character
- Convert each record into binary form
- Convert each binary record into equvalent decimal number X
- Find residues (x_1, x_2, …, x_{h+r})

Step6: End

Figure 4.4: Algorithm of Encoding by RRNS phase

4.3.2. Digital Signing Phase

After the residues are calculated, it must be signed and then spread to the storage server. The signing process is done by calls the *signature ()* function that takes the residues as argument as the format:

```
signatur (residues)
```

Inside the signature function many processes are performed as the following:

a) Set the global keys named (*p*, *q* and *h*), privet key (*x*) and per-message secret number (*k*) keys that are specified in Digital Signature Algorithm (DSA) as explained in chapter 3**.**
b) Calculating the remaining keys named *g* and public keys by applying Eq. (13) and (14) respectively.

c) Finding the multiplicative inverse (*MI*) of *k* by calling *multipleinverse ()* function, This function takes the *k* and *q* as requirements;

```
inv_k : multipleinverse (k ,q)
```

It works as algorithm that is shown in the Figure 4.5 (*k* refers to *num* and *q* refers to *mod*).

d) After that the hash value of the residues is found by calling *Hash_fun ()* function and apply the algorithm mentioned in the chapter 3 in Section (3.5.3). The hash function result return to *signature ()* function.

e) Using the (*p*, *q*, *g*, and *k*) with applying Eq. (15) to find the signature variable1 *r*, and using the (*inv_k*, *hm*, *x*, *q*, and *r*) with applying Eq. (16) in order to find the signature second variable *s*.

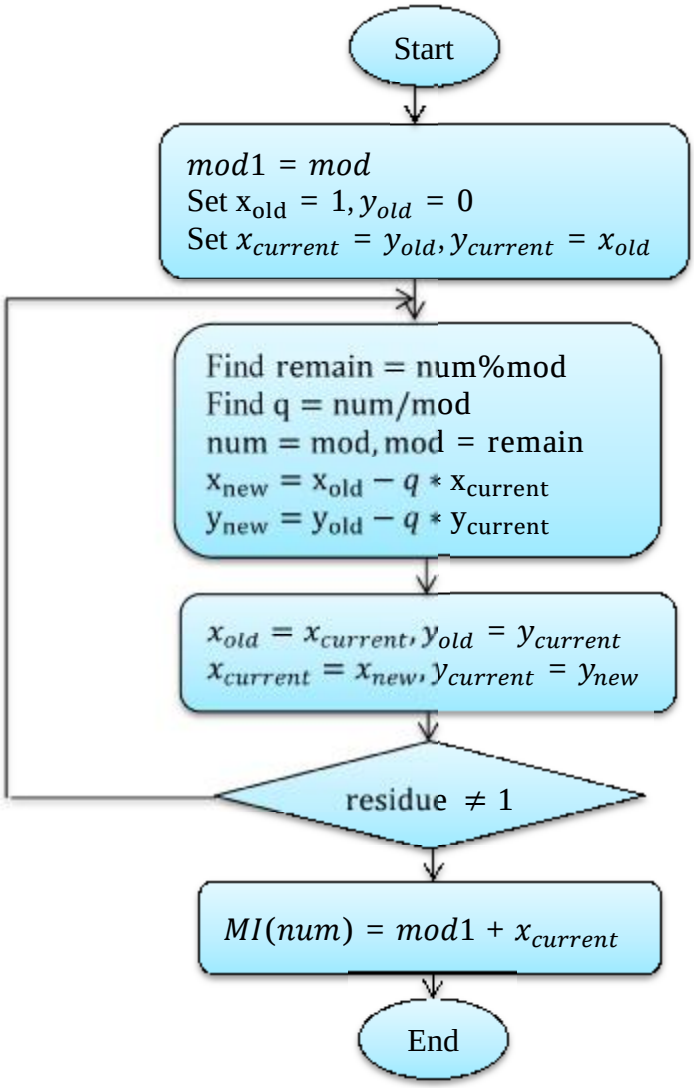


Figure 4.5: Flow Chart of Multiplicative Inverse Algorithm

4.3.3. Distribution of Residues Phase

This phase is responsible for distributing signed residues to set of trusted servers. Before the distribution process is star $h+r$ reliable storage server (SS_s) is chosen from a number of the servers. These SSs must be connected to the network that is created by PS. The IP of any connected server is kept in the program that is considered identification of that SS. Finite loop of the distribution process begins with first SS and end to last SS is happening.

The residues with the values of signature process is sent finally to the *sendmsg ()* function which is in turn spreading the signed residues to SS depending on the specified IP. The object in *Sendmsg ()* function is created from the *Udpclint* class is used to connect the program in PS with the program in SS through IP and the port number. Also the same object is used for sending the data after converting the data into array of bytes. Figure 4.6 is explained the distribution and signing processes.

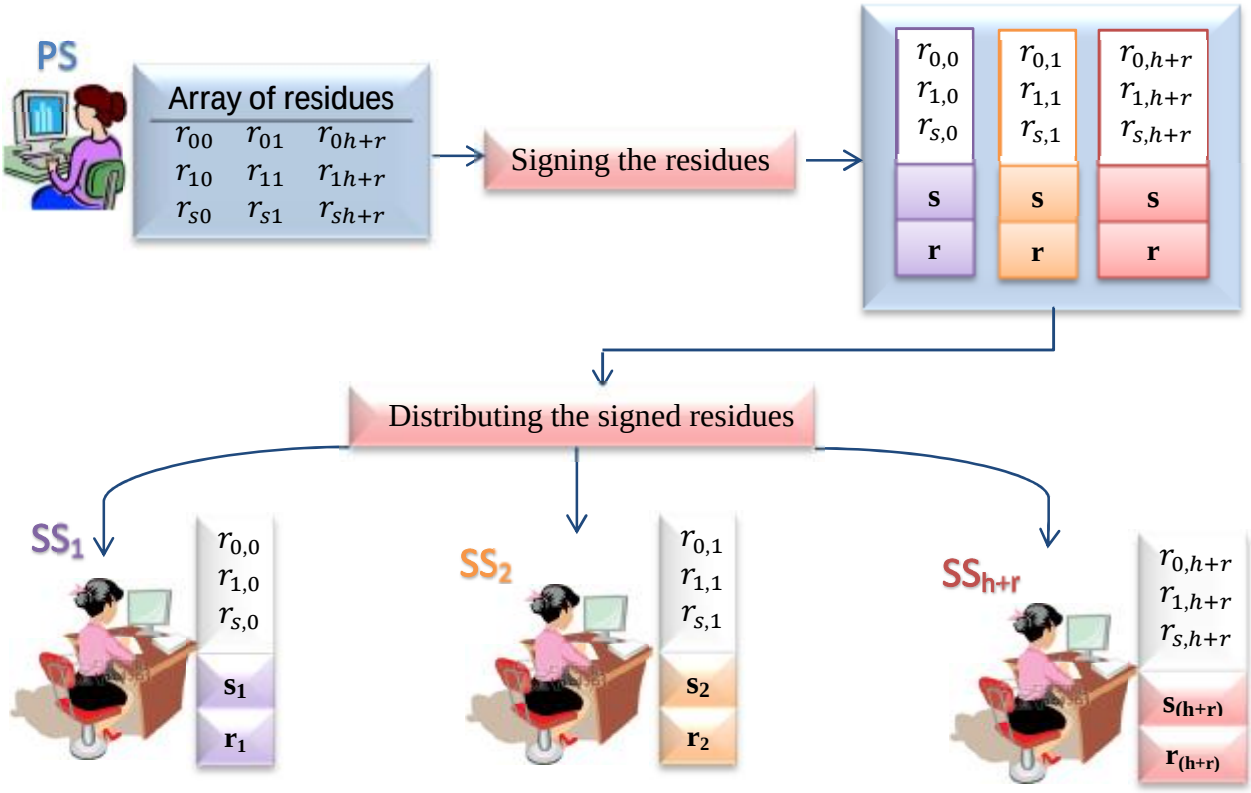


Figure 4.6: Flow Chart of Digital Signature and Distribution Process

4.3.4. Signature Verification and Authentication Phase

This phase is divided into two sub phases: verification, storage and authentication phase. The verification phase is happening when SS_s are receiving the signed residues from PS. The storage process is accrued when SS ensures that the residues are not modified during the transmission. Finally RSA algorithm is used for encrypting the storage locations, and the encrypted locations are sent back to the PS, this is done in the authentication phase. The details of each stage are explained below:

1) ***Verification phase:*** The process of receiving residues is made by using Thread structure that is allowed for the program to continue its process while it's receiving data. The *storage_Thread ()* function is built by creating object from *UdpClient* class and associated it with the number of the port. The address of the remote server (PS and RS) is given to the *RemoteIpEndPoint* object that is created from the IPEndPoint class. Byte array called *'receiveBytes'* is created and filled with data (residues) that are sent by the remote server. The address of the *RemoteIpEndPoint* object is checked if was from PS, the received data is placed in the *res_PS* string else was from RS is putted in *reder* string.

   SS by default knows the public parameters of the DSA (*p, q, g* and *public key (y)*). The verification processes are started by breaking the incoming message into the three parts: *residues*, *s* and, *r* (results of signature process). The residues are entered to the hash function to ensure that are not altered during transmission. Find the multiplicative inverse of value of *s*, the result value is used in Eq. (17) the result of this Eq, and the value of hash function is entered into Eq. (18). The value of *r* with Eq. (17) result are used in Eq. (19). The result of all above equations with *y* are used in Eq. (20) to find the verification

variable. This variable is compared with the *r*, if there is equal, the message is not modified.

2) ***Storage and authentication:*** This sub phase is responsible for storage and achieve confidentiality and authentication at the same time by first calling the *storage_encryption ()* function written in the below format

`storage_encryption (residues)`

Each residue is stored with identification (ID) of its location as the Table 4.5 bellow.

Table 4.5: Form of Storage

ID_1	$Residue_1$
ID_2	$Residue_2$
ID_s	$Residue_s$

The ID_s must sent to the PS in order to store it in the File Descriptor (FD). To enforce the confidentiality and ensure the authentication, RSA algorithm is used. IDs are encrypted first with the privet key of SS by applying Eq. (11), and then the encrypted result is encrypted again with the public key of the PS by applying the same equation

E (E (IDs, PR_{SS}), PU_{PS}).

Encrypted IDs are sent back to the PS using *sendmsg ()* function explained above.

The abstract of the above functions is explained in Figure 4.7 below. These processes are repeated for every SS where $s * (h + r)$ IDs are arrived to the PS.

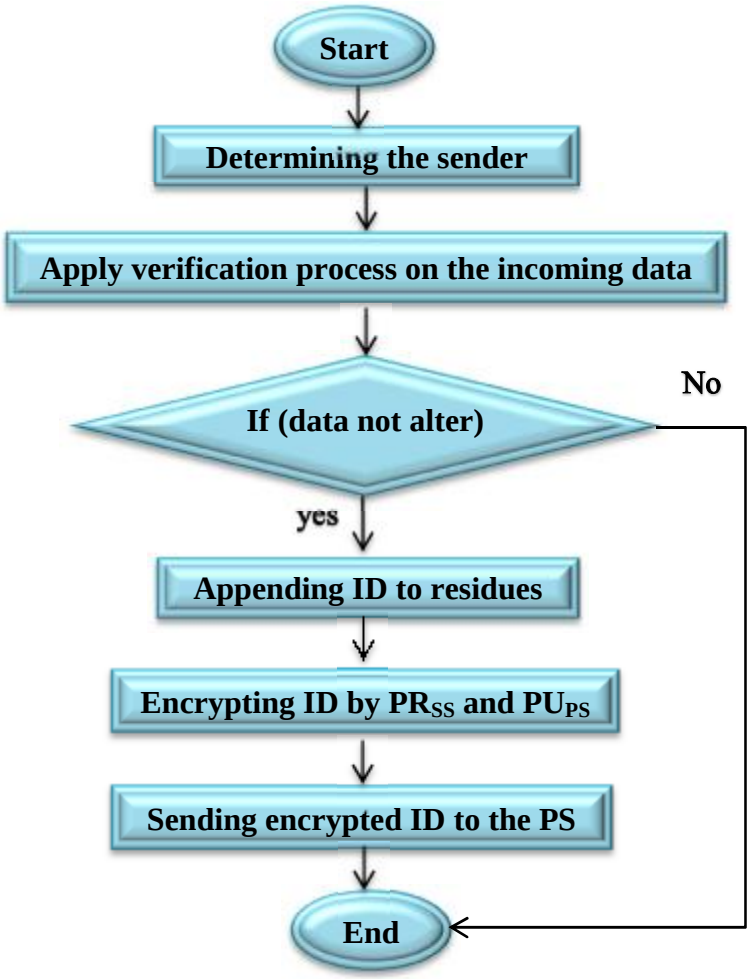


Figure 4.7: Flow Chart of Verification and Encryption algorithm

4.3.5. Encrypted File Descriptor Sharing

This phase is considered the core of the recovering original information on RS, after spreading the residues to set of reliable SS_s since the RS couldn't read the data before sharing the file descriptor (FD).

This phase is divided into many sub phases as shown in Figure 4.8: First dealing with ID included receiving and decrypting the IDs, and the second sub phase dealing with FD included creating, signing, encrypting, and sending the FD.

Algorithm Name: File Descriptor Sharing
In put: Received Storage locations (IDs)
Output: Sending Encrypted and signed File descriptor

Step 1: Start
Step 2: PS receives Encrypted(($s.(h + r)$ IDs) from h+r SS_s
Step 3: For t=0 to h+r

- Decrypting IDs at PS as follow: **D (D (IDs[t], PR_{PS}), PU_{SS}[t])**
- Keep IDs

Step 4: End for
Step 5: Creating FD=[(random number, modules, *h*, *r*, and *byte*s), IP_s, and ID_s]
Step 6: Signing FD
Step 7: Encrypting FD: E (E (FD, PR_{PS}), PU_{RS})
Step 8: Sending FD: PS send [E(FD), ss1,rr1]
Step 9: End

Figure 4.8: Encrypted file descriptor sharing algorithm

The entire sub phases are explained below in details.

1) Sub phases on The Identifiers

a. ***Receiving:*** The receiving process in the PS is similar somewhat these are explained in the SS. PS is received IDs from $(h+r)$ SS_s. The received IDs are stored in a 2D array. The first position of this array is filled with IDs that are received from first SS; the second position is filled from second SS and so on. Where finally 2D array with size $(s.(h + r))$ is occupied with IDs that are represented the positions of storage residues.

b. ***Decryption the IDs:*** The decryption operation is done during the *DEC_RSA ()* function. PS is deciphered the received ID_s first with its privet key that is calculated by applying Eq. (10) and then decrypted the result with public key of SS. Decoding the incoming message to recover the IDs is based on applying Eq. (12). The encrypted IDs are kept to include it in the FD.

2) Sub phases on The File Descriptor

a. Creating FD: FD is created by calling the *creat_filedescriptor ()* function that takes no arguments. In order to create FD, *I/O Stream* Class is used. Object from *FileStream* is created. This Object responsible for creating file with a specific name on the computer for writing. The second object is made from *StreamWriter* that is used for the printing process by calling the *Write* function. The contents of FD are: *modules, added random number, the value of h, r, bytes,* FD is also contains the *IPs* of the SS_s with its *decrypted IDs*.

b. Signing FD: Any process happened to the FD requires reading the FD. *I/O Stream* Class also is used for this purpose. Two objects are created: one from *FileStream* and the other created from the *StreamReader*. The *object1* is opened file with a specific name for reading. The *object2* is read FD to the end and store the content in a string variable. Then the *sign_FD ()* function is called to apply the DSA algorithm on that string in order to sing it. The function is calculated at first the two signature parameters (*ss1*, *rr1*) of the FD by applying Eq. (15) and Eq. (16). The explanations of the DSA algorithm are shown above.

c. Encrypting FD: After FD is signed, it must be encrypted. To encrypt FD, the string variable that is collected the content of the FD is converted to the numeric array. The function *Encrypted_FD ()* is called in order to encrypt FD with RSA algorithm $\mathbf{E\ (E\ (FD,\ PR_{PS}),\ PU_{RS})}$.

FD is encrypted first with the privet key of PS, and then it encrypted with public key of RS that is wanted to share the FD .

The result is returned to the *sign_FD ()*. The format of this function is

Encrypted_FD: ENC_RSA (File Descriptor, pub_reader1, n_reader1)

d. Sending FD: After FD is signed and encrypted is become ready to send to the RS depending on the IP of RS that is kept in PS. To send this FD, the *sendmsg ()* function is called, this function is explained earlier when it needed to send signed residues.

4.3.6. Information Reading Phase

Reading information process works by many steps. These steps are distributed among the RS and SS.

- **Processes on The Reader Server**

The process for reading patient's information is started by operating the *reader_Thread ()* function that has first received encrypted and signed FD with signature variables (ss1 and rr1) from PS.

FD is gone to *DEC_RSA ()* function that is decrypted the FD with RS's privet key and then with public key of PS **D (D (FD, PR_{RS}), PU_{PS})** to ensure the confidentiality and authentication. The format of this function is:

DEC_RSA (FD)

The verification process is achieved on the decrypted FD to ensure the integrity by applying the Eq. (17), (18), (19), and Eq. (20). The verification value *v* is resulted finally. Comparison between the two values (*v* and *rr*1) if there is equal, the message is not modified and the isolation process on the FD contents is started. The above explanation is shown in Figure 4.9.

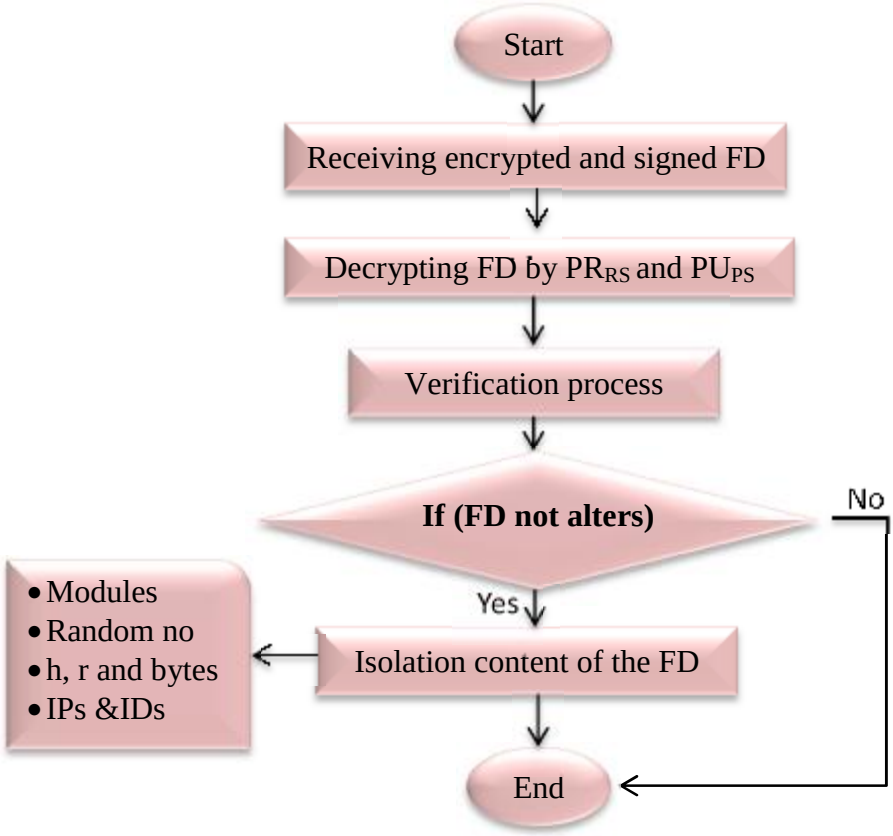


Figure 4.9: Flow Chart of Decrypting and Verification FD

In order to complete the reading process, the extracted IDs from FD must send to the SS to obtain the equivalent residues. The sending process must be secure so the IDs are encrypted with the privet key of the RS and then with public key of the SS using RSA algorithm. Finite Loop of sending process based on the number of the extracted IPs is happened. For encrypting the IDs for the specific IP, *ENC_RSA ()* function is called. This function takes the arguments: IDs and public keys of SS **E (E (IDs, PR_{RS}), PU_{SS})**. The format of this function is:

```
rsa: ENC_RSA (IDs, public key SS, public key n)
```

The *sendmsg ()* function is called to send the encrypted ID to the $(h+r)$ SS_s. The algorithm in the Figure 4.10 explains the above process.

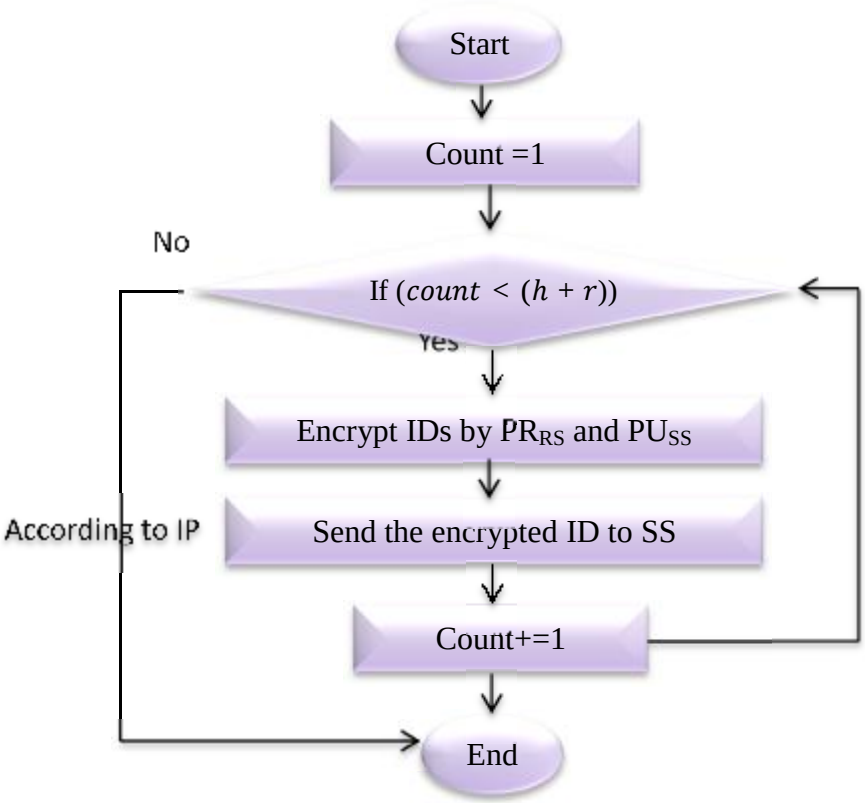


Figure 4.10: Flow Chart of Sending Process Algorithm

- **Processes on the Storage Server**

All SSs received the identifiers that are specified for the locations of storing residues. These IDs are encrypted by RS already. Therefor decryption process is done, *decryp_ID ()* function is made decryption process. This function is taken the received IDs as parameter, SSs are decoded the encrypted ID with its private key and then with public key of RS **D (D (ID, PR_{SS}), PU_{RS})**, and the result is compared. If all received IDs equaled the local stored IDs, the function return the equivalent residues. The format of this function written below:

```
Residues: decryp_ID (received encrypted IDs)
```

Then these residues are sent back to the RS by using *sendmsg ()* function. Figure 4.11 shows the process above.

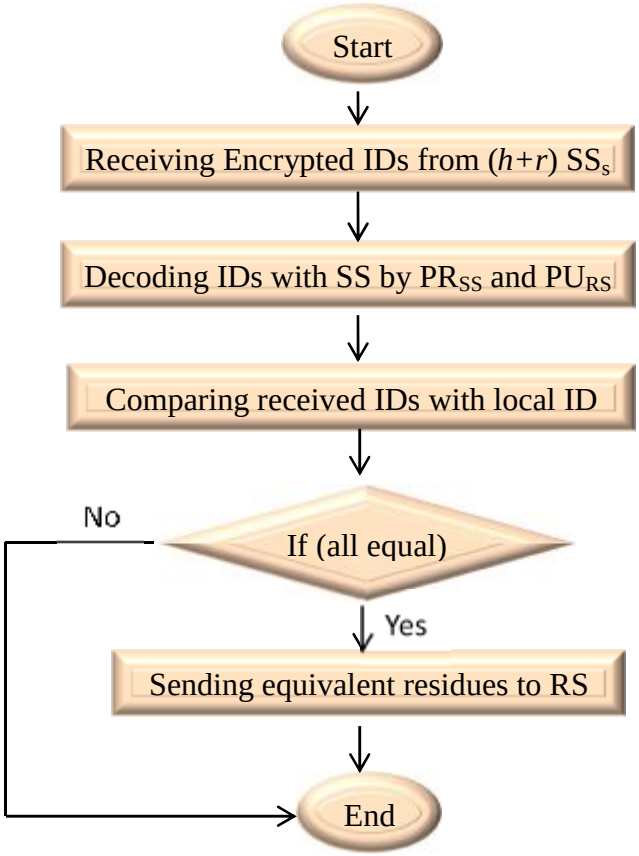


Figure 4.11: Flow Chart of Decrypting and Sending residues in SSs

- **Processes on The Reader Server**

The sent residues from SS are received at RS to enable it to recover the patient information. The steps for reconstructing information are:

Calling the *splite_residues ()* function that is taken the received residues as parameter as the follow format:

```
splite_residues(recive residues)
```

This function converts the residues to the big integer numbers. These residues result from RRSN algorithm. In order to recover the original data, these residues must be decoded. Three forms of the decoding process are produced in the designed system. These forms are described in the following Section.

4.4. Information Decoding

Three main techniques have been used for decoding the information, as described below.

4.4.1. Using Chinese Remainder Theorem (CRT)

The *CRT ()* function is called inside the *splite_residues ()* to decode the residues. The format of this function is: early

```
CRT (residues)
```

Inside this function, rang of moduli are calculated by applying Eq. (1). These moduli are obtained from FD as explained early. Apply Eq. (5) is enabled to recover the original big integer number.

The algorithm of CRT is illustrated in Figure 4.12.

Algorithm Name: CRT
Input: moduli, h, residues, and random number.
Output: Arr_original numbers

Step 1: Start:
Step 2: Range=1
Step 3: For k=0 to h
- Range=Range* moduli[k]

Step 4: End for
Step 5: For i=0 to i<number of records
- X=0
- For j=0 to j<h
  - mi =Rang/moduli[j]
  - X=(X+residues[i,j]*mi*MI(m_i,moduli[j]))%rang
- End for

Step 6: End for
Step 7: End

4.12: CRT algorithm

4.4.2. Base extension (BEX) with Mixed Radix Conversion (MRC)

The second decoding scheme is BEX with MRC. The initial steps mention above are remain same in this method until calling the decoding algorithm, where to recover the patient data, the *MRC_BEX ()* function is called to do this process. This function takes the received residues and modules as arguments as the below format:

```
MRC_BEX (residues, keys)
```

This function is taken *h* received residues and sent it to the *rdix ()* function is found the parameters of Eq. (6). To recover the original number, *original_num ()* function is called that is the function is applied Eq. (6). The Figure 4.13 shows the *MRC_BEX* algorithm.

Algorithm Name: BEX_MRC
Input: moduli, h, residues, and random number
Output: Arr_original numbers

Step 1: Start:
Step 2: For i=0 to i<number of records
- a[0]=residue[0]
- For j=1 to j<h
  - x=(residue[j]-a[0])*MI(moduli[0],moduli[j])
  - For z=1 to z<j
    - x = (x - a[z]) * MI(moduli [z], moduli [j])
  - a[z]=x% moduli[j]
  - End for
- Num=0
- For ii=0 to ii< length a
  - S=1
  - For jj=0 to jj<ii
    - S=S*moduli[jj]
  - Num =Num+ a[ii]*S
- End for

Step 3: End for
Step 4: End

Figure 4.13: MRC_BEX algorithm

4.4.3. New Chinese Remainder Theorem (CRT I)

The third way can be used for decoding RRNS technique is CRT I. As said above the initial operations at RS is the same for each decoding algorithm except the structure of CRT I. RS is built *CRT_I()* function specified for this theorem this function takes the received residues as arguments as the below format:

```
CRT_I(residues)
```

This function computes at first the array of k_s according to Eq. (9), and then it applies the Eq. (8) to recover the original weighted number as shown in the algorithm 4.14.

Algorithm Name: CRT I
Input: moduli, h, residues, and random number
Output: Arr_origenal numbers

Step 1: Start:
Step 2: Range=1
Step 3: For z=0 to h
- Range=Range* moduli[z]

Step 4: For z=0 to z<h-1
- M1=1
- M2=1
- For j=0 to j<=z
  - M1=M1*moduli[j]
- For u=z+1 to u<h
  - M2=M2*moduli[u]
- K[z]=MI(M1,M2)

Step 5: End For
Step 6: For gg = 0 to gg < number of records
- X = 0;
- For count = 0 to count < h
  - X = X + residus[gg,count];
  - count++;
  - For i = 0; i < h-1
    - mul3 = 1;
    - For o = 0 to o <= i
      - mul3 = mul3 * moduli [o];
    - mk = k[i] * mul3;

Next

- X = X +(mk * (residus[gg, count] - residus[gg, count - 1]));
- count++
- End For

- End for
- Arr_origenal numbers [gg] = X % Range

Step 7: End for
Step 8:End

4.14: CRT I Algorithm

After original numbers are generated, the random number that is added early is subtracted from these numbers to recover the decimal numbers, and then these decimal numbers are reconverted to binary form by called *conv_to_bin ()* function. The format of this function is:

```
conv_to_bin (decimal numbers)
```

From these binary numbers, the original characters are obtained by calling the *recoverchar ()* as the below format:

```
recoverchar (binary numbers)
```

The results of this function are ASCII codes of the original characters. After all that, the information is added to the DB using *OleDbConnection* class and putted in the record specified for each patient.

Chapter Five

Results and Discussion

Chapter Five

Results and Discussion

5.1. Introduction

In this chapter, the detailed results of simulation will be discussed and some of graphical user interface of the prototype system will be shown. The aim of this chapter is to give all detailed specifications for achieving distributed storage system and in the same time keep the security and privacy of the information. These specifications allow anyone who is interested in this domain to implement system easily. The system is implemented as four servers: one PS, two SSs, and one RS. More than these numbers can be used in real life. Each server has specified software and GUI. The feature of each server is summarized in the Table 5.1.

Table 5.1: Servers' Specification

Server name	Operating System	System Manufacturer	Processor	Memory
PS	Windows 7 Ultimate 32-bit	Dell Inc.	Pentium(R) Dual-Core CPU, ~2.2GHz	3072MB
SS1	Windows 7 Ultimate 32-bit	Hewlett-Packard	AMD Sempron(tm) , ~2.0GHz	2048MB
SS2	Windows 7 Ultimate 32-bit	FUJITSU	Intel(R) Core(TM)2 Duo CPU,~2.5GHz	2048MB
RS	Windows 7 Ultimate 32-bit	Dell Inc.	Intel(R) Pentium(R) CPU, ~2.2GHz	2048MB

5.2. Distributed Data Security and Privacy Interfaces

In order to ensure the security and privacy in the distributed storage system, many processes at different and scattered servers are done such as encoding, signing, distributing, storing, and decoding. The GUI of each server is designed to show the implementation of these operations. Before that some proper network settings must be set as shown in the appendix part.

5.2.1. PS GUI

PS is the core of the system because it has the main DB, prepare the local network to connect other servers and, also it creates the FD that is the most important entity in the reading process. The GUI contains four different groups. Each group contains many text boxes, buttons, labels, and combo boxes as shown in Figure 5.1. First group contains the *patient information* that is stored in the DB, it contains the following information:

- ***PN***: This field represents the patient number that is registered in the DB. It takes a number data type.
- ***Patient name***: This is the name of the patient, it takes string data.
- ***Age***: This field is numeric value represents the age of the patient.
- ***Gender***: The gender of the registered patient. It is a string value.
- ***City***: This field contains the name of the patient's city. It is a string value
- ***Drug name***: The name of the drug is entered in this filed. It is a string value.
- ***Doctor name***: It is string value represents the name of the doctor.
- ***Room No.***: It is numeric value contains the number of the room.
- ***Date***: This field contains the date of entering the patient to the hospital.

All these boxes have the read only property to prevent anyone to modify the data. All these above fields are filled when the program is run as a result of connection DB to the program software. The information about the patient is entered in the MS Access database not directly through a running program.

- The ***Show DB*** button displays DB's form that is containing the whole DB, It includes:
  - ***DataGridView***: This is tool specified for displaying the entire DB.
- ***Last***: This button displays the last record in the DB.

- ***First***: This button shows the first record in the DB.
- ***Next***: The next record is displayed by this button.
- ***Pre***: This shows the previous record.

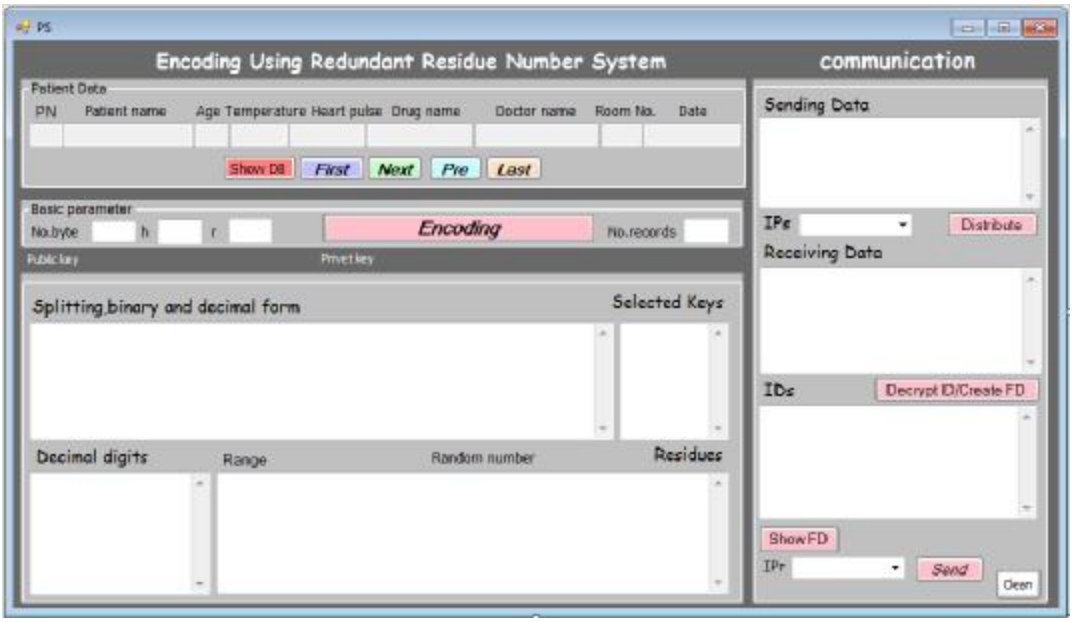


Figure 5.1: PS's GUI

The second group contains the *basic parameters* of the RRNS methods. It includes:

- ***No.byte***: This represents the number of the characters that the user based on it when he/she split the patient's data, is set by the user and it must be a numeric value.
- The ***h*** and ***r***: These numeric values are filled by the user, *h* represents the non-redundant while *r* represents redundant value.
- ***Encoding***: This button refers to the beginning of encoding process by using all data that are mentioned in the groups above.
- ***No. records***: This field represents the number of records that are calculated within an encoding process by using the Eq. (20).

The third group in the PS's GUI has the results of the *encoding process* that includes the following information:

- ***Splitting, binary and decimal form***: This field represents the first level of RRNS technique. Where the data at first is split according to the

basic parameter to generated number of records, then these records are converted to binary form and finally the decimal numbers are calculated. These processes are shown in this text box.

- ***Selected keys***: This field has the modules that are selected from built library and will be used in the encoding process.
- ***Range***: This label represents the range of the keys. This range control of the encoding where if the range was less than the $2^{\text{no. Bytes}}$, a message box will appear to require from the user to increase the number of the keys or decrease the No.byte.
- ***Random number***: This displays the generated random number that added to the decimal numbers.
- ***Decimal digits***: This box shows the decimal numbers after adding random number. The decimal number must be less than the range.
- ***Residues***: The result of the RRNS method is producing the residues those are displayed in this box.

Final group is specified for sending and receiving processes. Communication group consists of:

- ***Sending data***: This box shows any sending data process either the signed residues that are sent to the SSs or the signed and encrypted FD that is sent to the RSs.
- ***IPs***: This contains the IPs address of the SSs.
- ***Distribute***: This button is specified for distributing the signed residues to the SSs. It will send all residues at the same time to the all determined SSs.
- ***Receiving data***: Any received data is displayed in this box that is the encrypted IDs that is sent by SSs.
- ***Decrypted ID***: This button is responsible for decrypting the received ID and creating FD.

- ***IDs***: This box shows the decrypted IDs after storing in the PS.
- ***Show FD***: PS after receiving the IDs, it will begin to create the FD that contains the mentioned information in the chapter 3. This button displays FD by opening its form that contains:
  - ***File descriptor***: This text box shows the contents of the created FD.
- ***IPr***: The IP address of the RSs is displayed in this box.
- ***Send FD***: This button according to the determined IPr, it sends the encrypted and singed FD.
- ***Clean***: This button responsible for cleaning all GUI.

In the PS's form, public and the private keys of the PS are also shown.

5.2.2. SS's GUI

PS send the residues to the number of SSs in order to store it. SS performs many processes on these residues. The GUI is the same for all SSs. It consists of group boxes, text boxes, labels, buttons and a combo box as shown in Figure 5.2. GUI deals with the following process:

- ***Receive data from servers***: SS receives signed residues from PS and/or encrypted IDs from RS.
- ***Residues***: After splitting the received data into residues and signature parameters. The resulted residues are putted in this box.
- ***s'*** and ***r'***: These boxes display the signature values that are split from received data.
- ***v***: This box shows the value of the verification process.
- ***Verify/storing/Encryption***: This button is responsible for doing the verification process and put the result in the v' textbox. If the v' equal the r' value, the SS will add the ID of the storing location for the received residues, these IDs are encrypted and sent to the PS in order to

keep it in the FD. Otherwise the message box will show that the message is altered during transmission.

- ***Associate ID with residues***: Inside this text box the ID of the storage location is displayed with stored residues.
- ***IP***: This shows two IP addresses of the (PS and RS) servers.
- ***Send***: For sending data, send button is used.
- ***Sending data***: This textbox shows the data that are sent to PS that is includes the encrypted IDs or the residues that will be sent to the RS.
- ***Public key/private key***: These labels display the public and private key for SS.

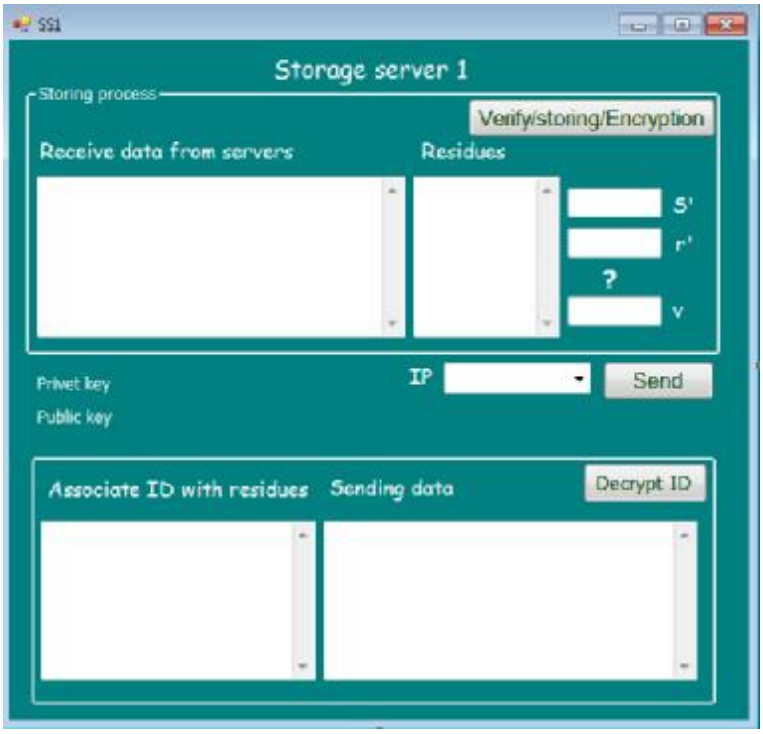


Figure 5.2: SS's GUI

5.2.3. RS's GUI

RS is the server that is residing remotely from PR and SSs and it needs to read the patient's data that are distributed among the SSs. In order the system to be tolerant, more than one RS is used. The RS GUI contains two groups as shown in the Figure 5.3; the first one consists of the following tools:

- ***Received data***: It is responsible for displaying any received data whether the encrypted and signed FD from PS or the residues from SSs.
- ***Verify FD***: This button has made the verification process on the signed FD after decrypting it.
- ***s*** and ***r***: These textboxes show the extracted signature values after received FD.
- ***v***: The result of the verification process is put in this textbox. If the two values (r and v) are equal, the received FD is displayed in a new form called FD. Otherwise the message box will be shown that told the received message is modified during transmission.
- ***FD***: This button is responsible for displaying the content of the FD after receiving it from PS. It contains:
  - ***File descriptor***: Textbox shows stored FD.
- ***Encrypted ID***: The Received ID must be encrypted before sending it. These encrypted IDs are displayed in this text box.
- ***Send ID***: Is the button for sending the encrypted IDs to the SSs.

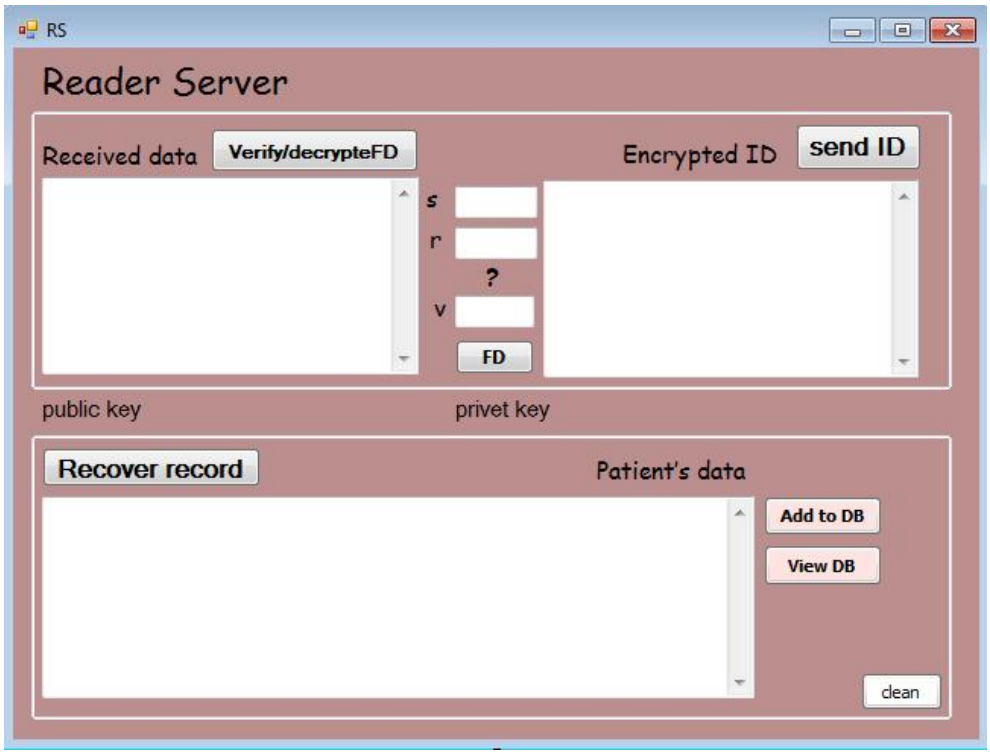


Figure 5.3: RS's GUI

- ***Public key and private key***: Two labels show the public and the private key of the RS.

The second group box is Recover record group. It consists of the tools below:

- ***Recover record***: After sending the ID to the SSs, SSs send the equivalent residues back to the RS. This button calls the reading processes on the received residues to recover the patient's data.
- ***Patient's data***: The processes for recovering data will be displayed in this text box that is including the decrypted residues.
- ***Add to DB***: This button is responsible for adding the recovered records to the DB.
- ***View DB***: This button is opening DB form that contains the following tools (as shown in the Figure 5.4):
  - ***dataGridView:*** This tool shows the DB after adding the patent information.
  - ***Enter the ID of record***: Inside this textbox, the ID of the record that is required to delete is entered.
  - ***Delete***: This button executes the record deletion process.
- ***Clean***: This button is cleaning all GUI.

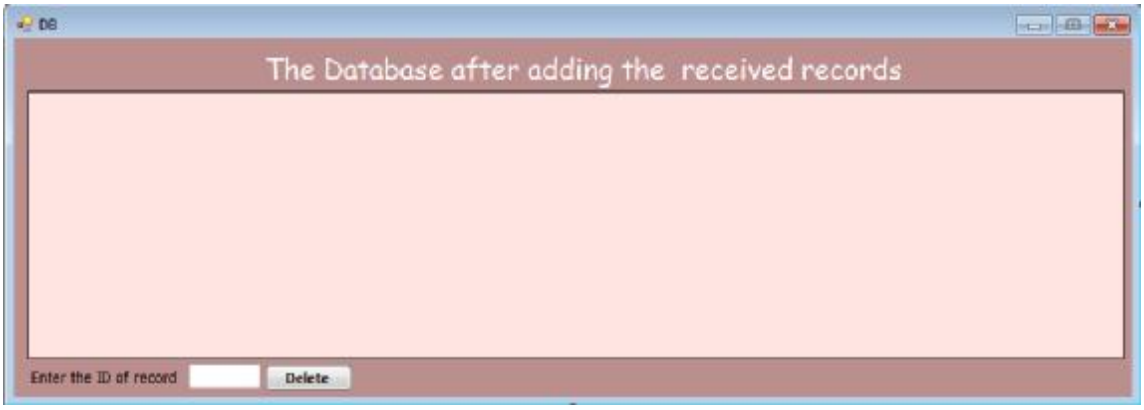


Figure 5.4: Database's GUI after Adding Record

5.3. Run of The System

When the simulated prototype system is running, more windows are used. The basic windows are PS, SS and RS. Other windows such as DB and FD are used to understand the work. The following two remarks need to be noted:

Remark1*:* At PS any sent data will appear in the *sending data* textbox and any receiving data is shown in the *Receiving data* text box. At SSs any sent data will appear in the *sending data* textbox and the received data is appearing in the *Receive data from server* text box. At the RS, The *Encrypted ID* text box is responsible for sending the data and the *Received data* textbox shows the received data.

Remark2: The color of any sending text box is becoming Powder Blue color and the color of any receiving textbox is converted to the red color.

The first step will happen when the program is run; the DB will connect with the system and when PS press on the *Show DB* button, the built BD will appear as shown in the Figure 5.5. The residues generation process is started by setting the *no. byte*, *h* and *r* values by the PS. When the *Encoding* button is pressed, the ciphering process is starting and the residues will calculated.

After that, the *Distribute* button is pressed in order to distribute the residues to $h+r$ SSs according to the IPs addresses that are specified in the program later. Before the residues are sent, PS will sign it's to ensure the authentication and integrity. The results of encoding, signing and distribution process are shown in the Figure 5.6.

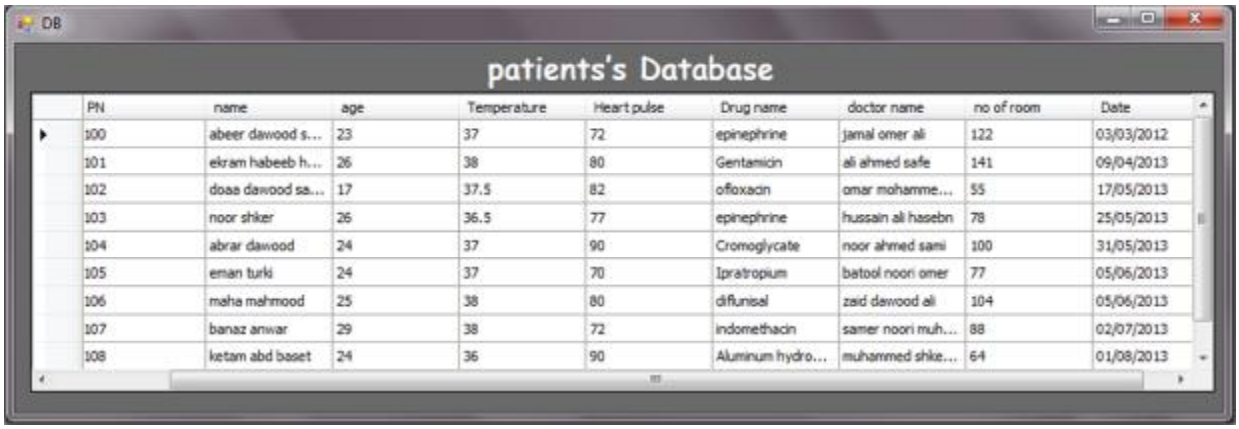


Figure 5.5: Result of connection DB with main program

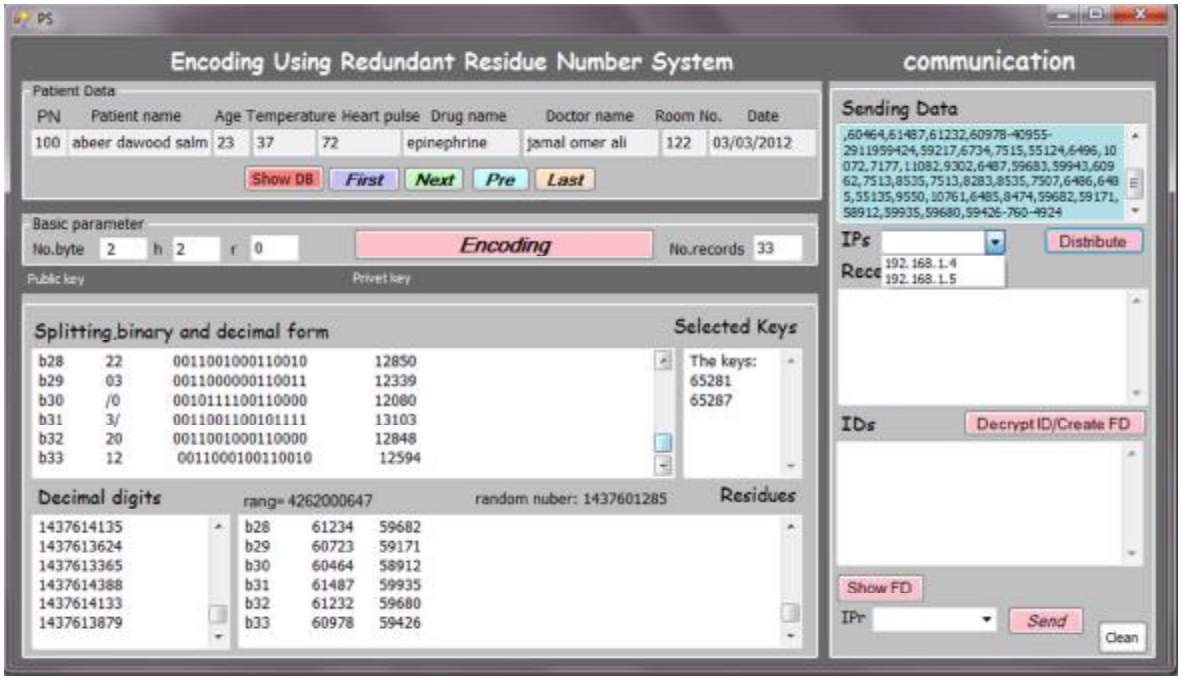


Figure 5.6: Encoding, Signing and Distributing Process

SS will receive signed residues and when the SS is pressed on the *Verify/storing/Encryption* button, The SS verifies that these received residues are not altered during the transmission. If the result of the verification process is true, a message box will appear ensure this event as shown in Figure 5.7.

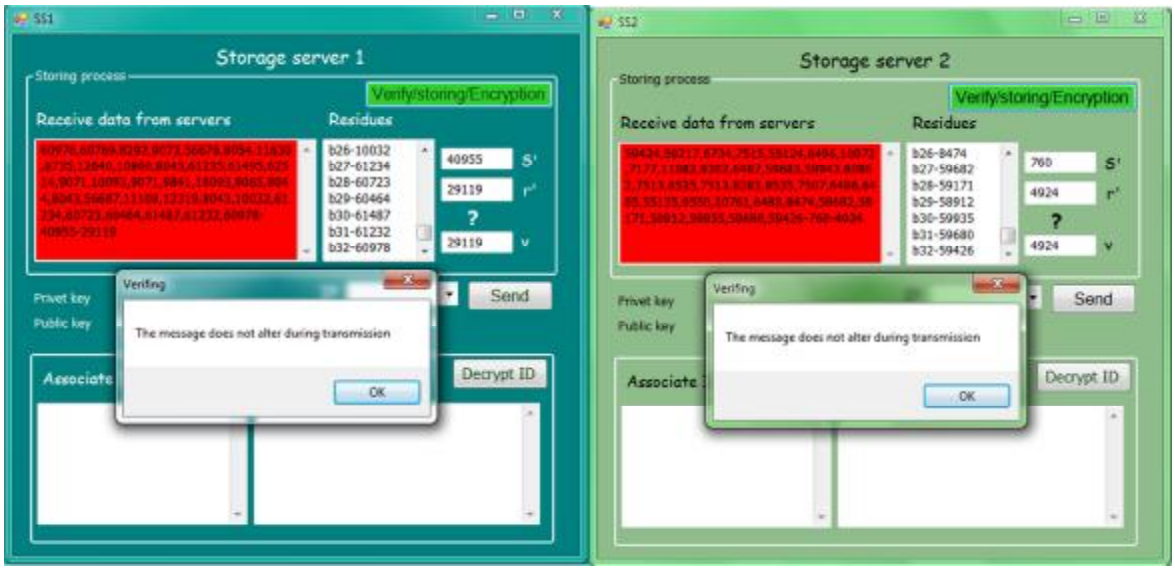


Figure 5.7: Verification Process

SS will store these residues and will add ID for the storage locations; these IDs shown in the *Associate ID with residues* textbox. SSs must send these IDs to the PS that is in turn store it in the FD therefore SSs determines the IP of the target server by using the *IP* combobox and sends the encrypted ID to the PS as shown in Figure 5.8.

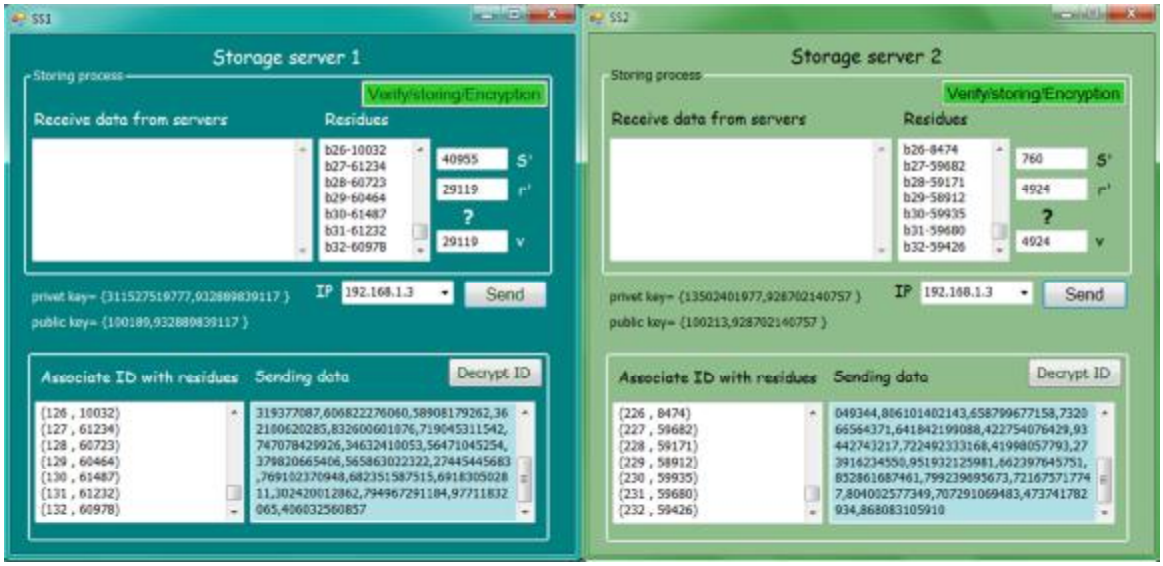


Figure 5.8: Storing and Sending Process

PS receives the encrypted ID and it must decrypt it. To do this we will press the *Decrypt ID/Create FD* button in order to first decrypt the received IDs from SSs and put the result of the receiving IDs in the *IDs* textbox and second

to create FD . FD can be shown by press on the *show FD* button as shown in the Figure 5.9.

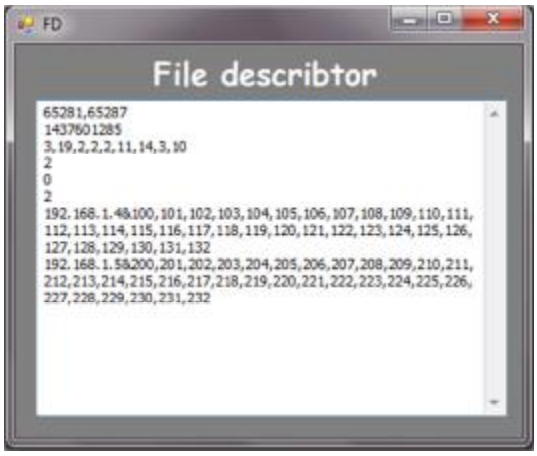


Figure 5.9: The content of the FD

Now there is RS wants to read the data of the patient, This RS must share the FD, therefore PS must send the FD to that RS by clicking the *Send* button after determines the IP of the RS by *IPr* combo box. This FD is signed and encrypted with the private key of PS and again with public key of the RS as illustrated in Figure 5.10.

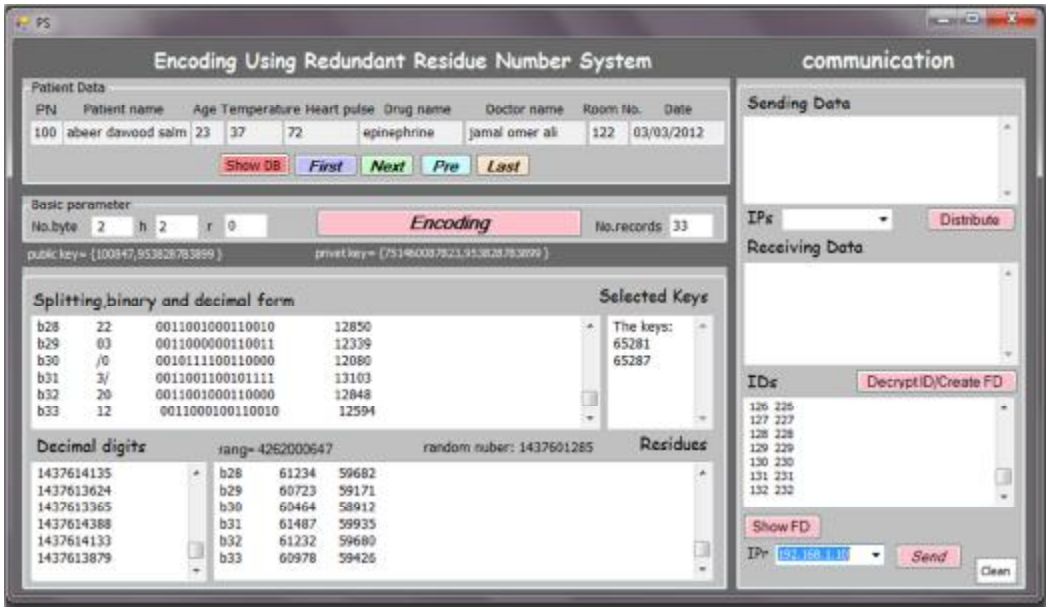


Figure 5.10: Sending FD

The procedure of receiving data at RS is similar to the receiving process in the PS and SS. Received data are putted in the *Received data* text box. Then RS must verify and decrypt FD by clicking on the *Verify/decrypte FD* button

where the signature values *s* and *r* are printed in the *s* and *r* textbox respectively. After that the verify value is displayed in the *v* textbox. If *v* equal *r* message box will appear to ensure that the FD is not modified during transmission as shown in the Figure 5.11.

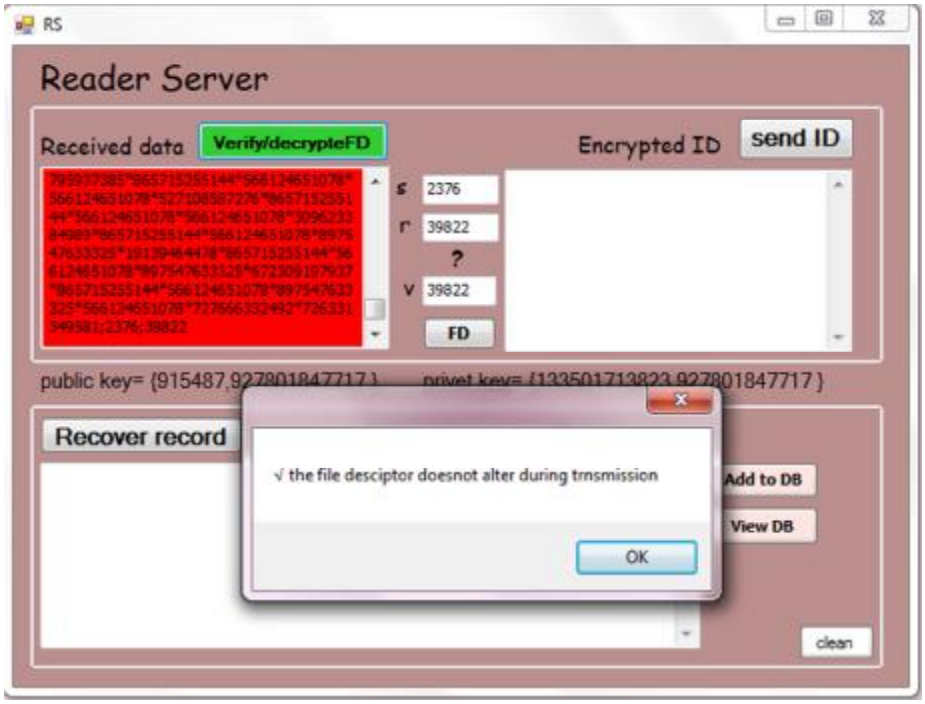


Figure 5.11: Receiving, verifying, and decrypting FD

In RS, we will press on the *FD* button to show the received content as illustrated in the Figure 5.12**.**

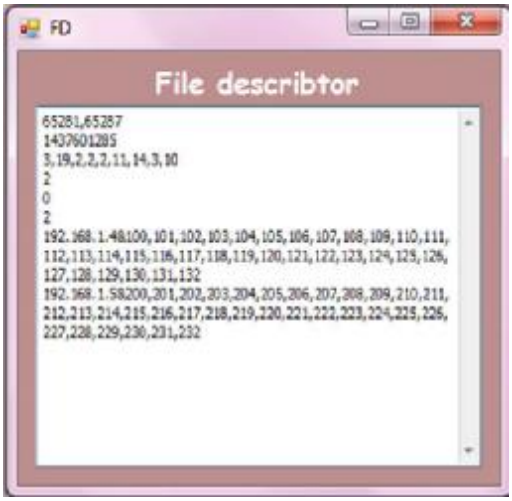


Figure 5.12: FD at RS

RS is used IPs of the SSs that is found in the FD and based on it, RS sends the associated IDs to the SSs. RS presses on the *Send ID* button to send these

ID to the SSs after encrypting it with RS's private key and with the public key of the SSs as shown in Figure 5.13.

Figure 5.13: Sending encrypted IDs

SSs are receiving encrypted IDs and decrypt it when clicking on the *Decrypt ID* button. SSs will test received IDs, a message box will appear to tell SS that the received IDs are similar to its own. Figure 5.14 illustrates this operation.

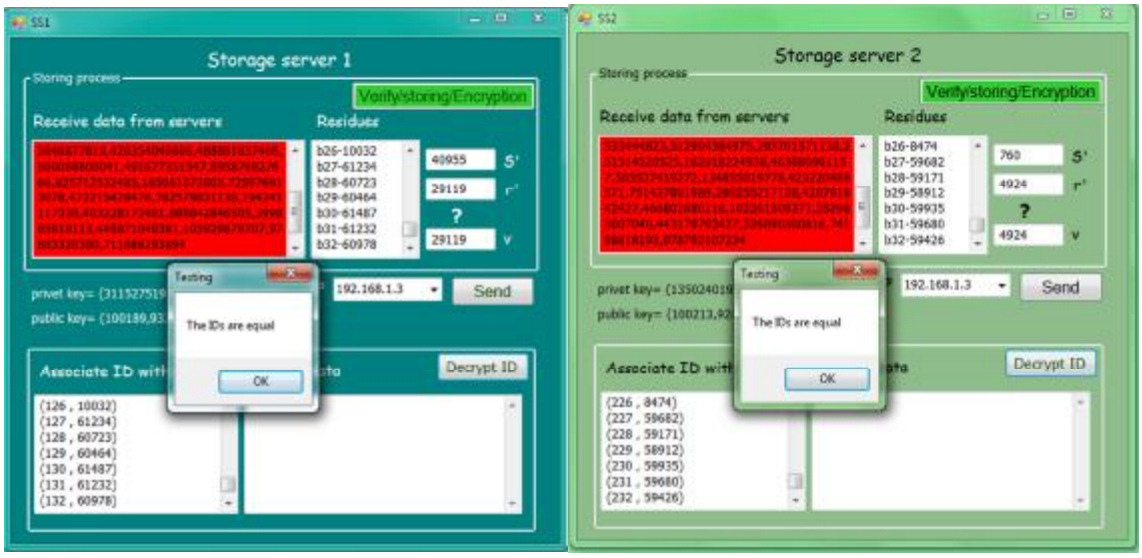


Figure 5.14: Received and Decrypted IDs

Then SSs will send the equivalent residues to the RS after determining the IP of that server. RS receives the residues that are sent by SSs and it will begin to recover the patient's data by pressing on the *Recover record* button as shown in the Figure 5.15.

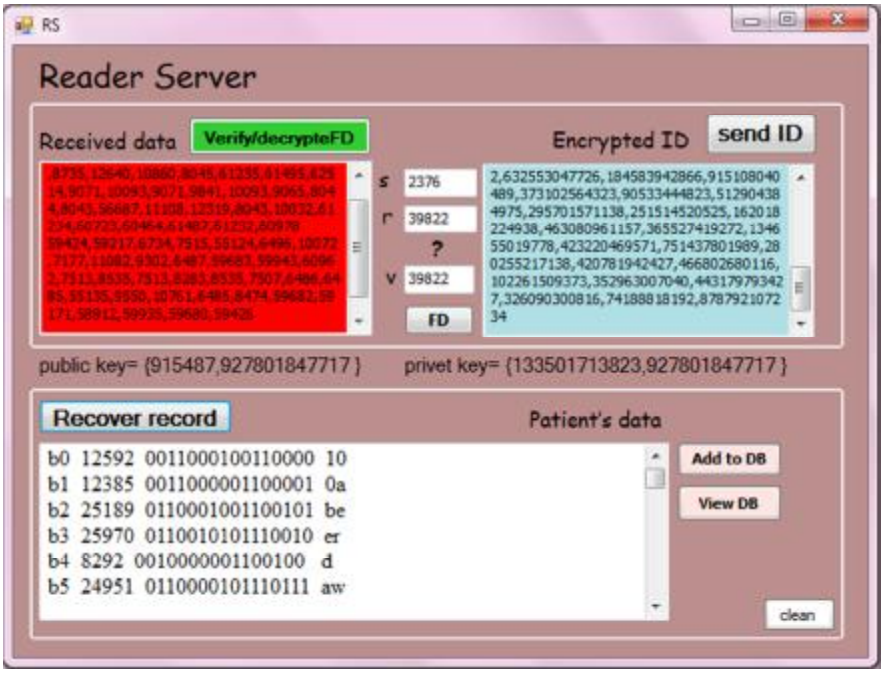


Figure 5.15: Recovering Patient information

RS must again store that recovered record in the DB by clicking on the *Add to DB* button. To show the DB after adding record RS press on the View DB as shown in the Figure 5.16**.**

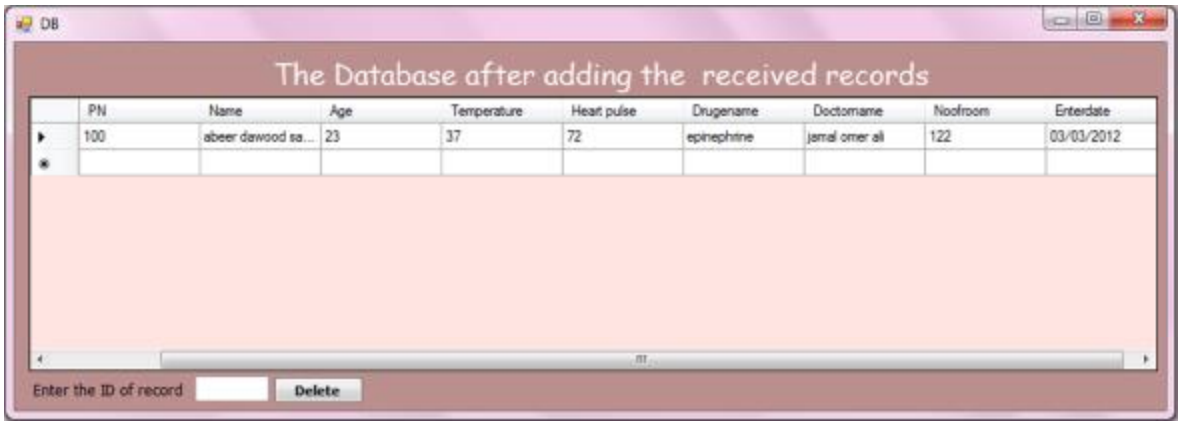


Figure 5.16: The DB after Adding the Record

To delete any record from DB the RS can do this by entering the ID on that record and press on the *delete* button.

5.4. Security Analysis

In this section, the security requirements that are achieved by the system are analyzed as follows:

5.4.1. System Availability and Dependability

The dependability is achieved when the PS uses the redundancy where the RS can reconstruct the data up to $s \leq r$ residue erasures. Such as if the value of $h = 4 \; and \; r = 2$ and the residues are ($x_{1.1}$, $x_{1.2}$, $x_{1.3}$, $x_{1.4}$, $x_{1.5}$, $x_{1.6}$) and if the RS is failed to receive 2 residues from SSs due to network problem, attack on some SS or any other event, decoding 4 residues ($x_{1.3}$, $x_{1.4}$, $x_{1.5}$, $x_{1.6}$) with equivalent moduli (m_1, m_2, m_3, m_4) will give legitimate decimal values that is enough to recover the patient's data because the erasures were less than the redundancy.

Now suppose that RS receives all the residues but with corrupt values, RS can reconstruct the records up to $\frac{r-s}{2}$ corrupted residues.

For example: take the same value of h and r above and suppose the RS receive ($x_{1.1}$, $x_{1.2}$, $x_{1.3}$, $x_{1.4}$, $x_{1.5}$, $x_{1.6}$) the value of $x_{1.2}$ is corrupted. Applying the reading algorithm on all received residues will give the illegitimate number in this case the error is detected. But when the RS applies reading algorithm on the ($x_{1.1}$, $x_{1.3}$, $x_{1.4}$, $x_{1.5}$, $x_{1.6}$) with moduli (m_1, m_3, m_4, m_5, m_6) the resulted value is legitimate and the record is decoded probably. But in the case arriving 2 residues with corrupted value the error can be detected but the record cannot be reconstructed.

5.4.2. System Confidentiality and Authentication

Because the system aims to distribute storage, this means the network communication is used, therefore confidentiality of communication must be ensured. The confidentiality of the proposed system is ensured at two levels of system implementation.

The first level was by using RRNS technique where in order to recover the patient information on the RS, the original modules (primes) must be known to allow of RS to decrypt the data. Therefore RS must share FD.

In the system RSA algorithm is used to achieve confidentiality, authentication and non-repudiation at the same time. Where the sender is encrypted the message with its privet key, the receiver can guarantee the identity of the sender, in this case authentication and non-repudiation are ensured. Then the sender encrypts the message again with the public key of the receiver, and thus the confidentiality is achieved and the message is protected from eavesdroppers. The cases of ensuring the confidentiality, authentication, and non-repudiation of the system are done in the following places:

- SSs and PS

  SS $\longrightarrow$ E (E (ID, PR_{SS}), PU_{PS}) / **PS** $\longrightarrow$ D (D (ID, PR_{PS}), PU_{SS})

- PS and RS.

  PS $\longrightarrow$ E (E (FD, PR_{PS}), PU_{RS}) / **RS** $\longrightarrow$ D (D (FD, PR_{RS}), PU_{PS})

- RS and SS

  RS $\longrightarrow$ E (E (ID, PR_{RS}), PU_{SS}) / **SS** $\longrightarrow$ D (D (ID, PR_{SS}), PU_{RS})

Public key cryptography algorithms are in general executing slower than Symmetric key cryptography because it must be able to publish the encryption key without disclose the decryption key. This requires heavier mathematics, compared to symmetric encryption which is "just" making a big tangle of bits. Most known asymmetric encryption systems seem to achieve the needed security, but at some relatively heavy computational cost. Public key algorithm can be used if the sent message is small in size as case of the proposed system.

In spite of ECC provides high security with small key size but the reason for choosing RSA because it's given acceptable security with less complexity

comparable to the ECC, and RSA is used in the system to encrypt storage locations of the residues that are already encrypted by RRNS.

5.4.3. System Integrity

Some information in the system is considered critical and using RRNS methods alone may be not enough, therefore the system uses the DSA to add signature to its data in order to confirm that the data not modified by an attacker during transmission and also ensure the authentication service. DSA is used by PS at two positions:

a. PS adds signature to the residues before sending it to the SSs.

b. DSA is used on the FD before encrypting and sending it to the RS.

The assurances of integrity are coming from using the SHA_1hash function algorithm that effect on the signature variables s and r ,if the residues or the FD changed during the transmission the verification process at the SSs or RS are effected where the hash value will change and the verification variable v will not equal the r value. The authentication service is achieved by DSA when the PS is signed the information with its private key. The SSs and RS verify the data with the PS's public keys.

All security services that are ensured by applying our proposed system can be abstracted as shown in the Table 5.2.

Table 5.2: Security services

Algorithm / ***services***	***RRNS***	***DSA***	***RSA***
Confidentiality	**yes**	**-**	**yes**
Authentication	**-**	**yes**	**yes**
Authorization	**-**	**-**	**yes**
Dependability	**yes**	**-**	**-**
Integrity	**-**	**yes**	**yes**
availability	**yes**	**-**	**-**
Non-repudiation	**-**	**yes**	**yes**

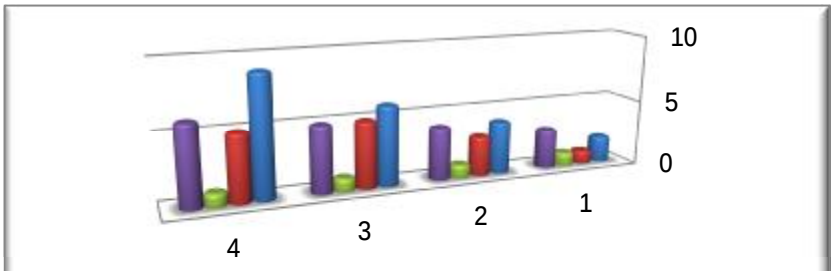
5.5. System Efficiency

In this section, the performance of proposed system is discussed in hand of code efficiency of RRNS with different moduli value and the algorithms that are used for decoding RRNS. The reason for this is to give indication on the best methods that are used to produce a secure and efficient system at the same time.

5.5.1. Code Efficiency of RRNS

The system uses number of moduli for the encoding process. The PS splits the data into records according to the value of moduli such as it can use 2 or 5 bytes. Suppose PS is using the modules from Table 3.2, the largest value of the moduli is 65536 that is equal to 2^{16}. This means PS can split file to every 16 bit (2 byte) only with at least h=2 because if h =1 the range M according to Eq. (1) will equal moduli that is error because 2^{16}>M, and this is not allowed in the RRNS, therefore the value of h must be greater than 1.

Code Efficiency of executing the system with 16 bits, and with using redundancy is shown in the Figure 5.17. CE is calculated by applying Eq. (4)

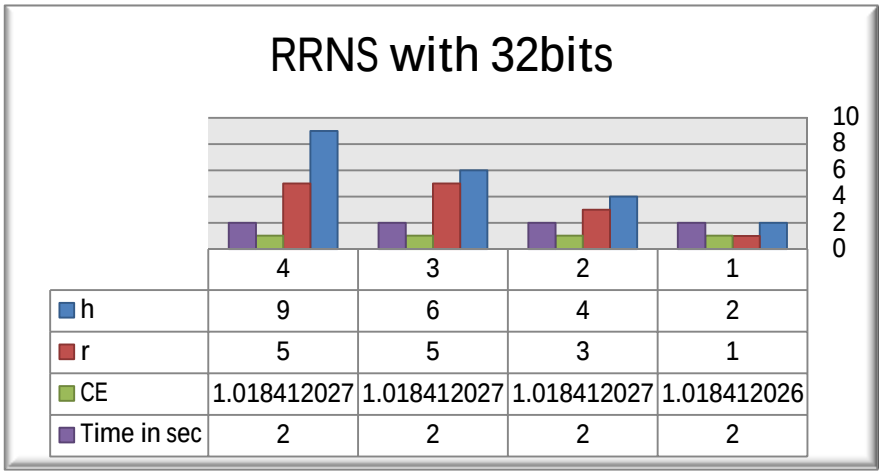


	4	3	2	1
h	9	6	4	2
r	5	5	3	1
CE	0.999712175	0.999698678	0.99967964	0.999656758
Time in sec	6	5	4	3

Figure 5.17: CE with redundancy and byte=2.

If the patient data was so much, splitting file characters to every 16 bits will produce large numbers of records and the time of the encoding will increase. Therefore PS can use the modules at Table 3.3 that can split the data to every

32 bits (5 bytes) because the largest value of these primes is 6461333947. The CE is calculated by applying same equation except $e = 32(h + r)$ as shown in the Figure 5.18. The results show that RRNS with big moduli value is given increasing CE and the time and the number of records is decreasing.



	4	3	2	1
h	9	6	4	2
r	5	5	3	1
CE	1.018412027	1.018412027	1.018412027	1.018412026
Time in sec	2	2	2	2

Figure 5.18: CE with redundancy and byte=5

5.5.2. The Performance of Decoding Algorithms

In this subsection, the decoding algorithms are evaluated. Where three theorems are applied for decoding RRNS that are: CRT, BEX with MRC, and CRT I. The performance of these algorithms is calculated depending on the following parameters:

1. Execution Time

The Time (in millisecond) of executing each algorithm is shown in the Figure 5.19 where CRT I is faster in reading information and recover the original decimal number. Even the PS encodes the data with large modules, CRT I remain executes with the least time.

2. Cyclomatic complexity (CC)

Software metric (measurement) is used to show the complexity of a program. This metric is based on a control flow representation of the program. Control flow describes a program as a graph which consists of nodes and edges. The difference between the above reading algorithms is

illustrated in the Figure 5.20. The CRT had lower value of Cyclomatic complexity. CC is measured by using Microsoft Visual C# tool as follow: Analyze→calculate code metrics for solution.

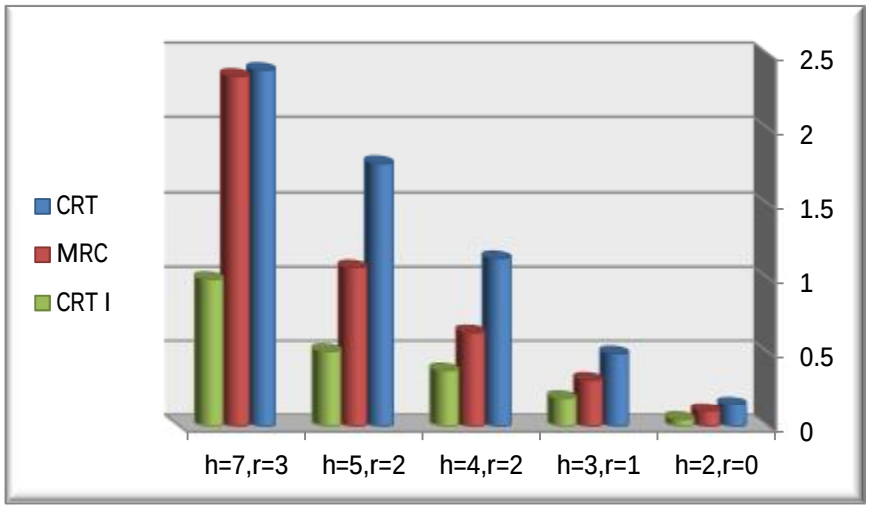


Figure 5.19: Time of the Decoding Algorithms

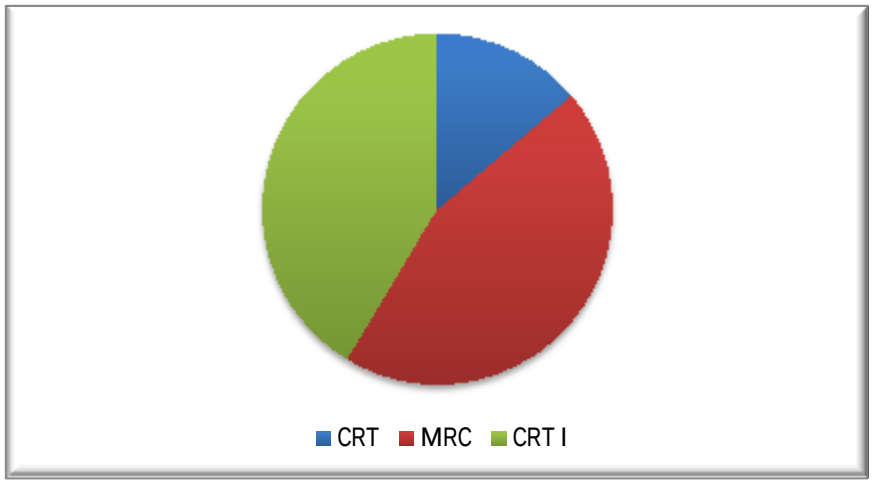


Figure 5.20: Cyclomatic complexity of reading algorithms

The implementation of CRT is computationally intensive for large moduli values as it deals with modular operations with a large value of M. Thus to avoid processing large numbers, the use of base extension (BEX) operation in integration with mixed radix conversion has been frequented. However, the Mixed Radix Conversion requires finding the mixed radix digits which is a slow process therefore New Chinese Remainder Theorems were designed to make the computations faster and efficient without any overheads.

Table 5.3 shows the times in milliseconds of encoding and decoding algorithms with varying number of (h+r) and byte to a query has size 66 characters. Whenever the number of the byte is increased the time of the encoding, time of decoding, and the number of the records is decreased. The more number of storage servers make the process of encoding and decoding long.

Table 5.3: Times in milliseconds of Encoding and Decoding algorithms

byte	no. rec	h+r	Time of Encoding	Time of CRT	Time of MRC	Time of CRT I
2	37	2+1	3307.1776	0.1586	0.0979	0.1105
2		4+2	4099.4608	1.1305	1.6863	0.3784
2		6+3	4282.9582	2.689	1.6513	0.8263
2		9+6	6216.5772	3.8943	4.8485	2.2793
5	15	2+1	2829.9167	0.3998	0.1241	0.0998
5		4+2	2984.1987	1.9807	0.635	0.383
5		6+3	3331.1054	1.7227	1.3872	0.8865
5		9+6	3833.5385	3.4132	3.1174	2.7628

5.5.3. The Performance of System

In this subsection, the performance of functions that ensure the security and privacy of the system are evaluated on the hand of time and Cyclomatic Complexity. This evaluation is based on the two Tables of moduli. The performance of the following operations is explained in the Figures below, and all times are calculated in millisecond:

- ***Signing and verifying of the residues***
- ***Storing, encrypting, and decrypting of storage locations***
- ***Creating FD***
- ***Signing, encrypting, verifying and decrypting FD***

Figure 5.21 shows the difference between the times of the signing of the residues at PS with verifying process on these residues at SSs.

Figure 5.22 explains the time of storing residues, encrypting the storage locations (IDs) after adding it to the stored residues, and decrypting IDs at PS.

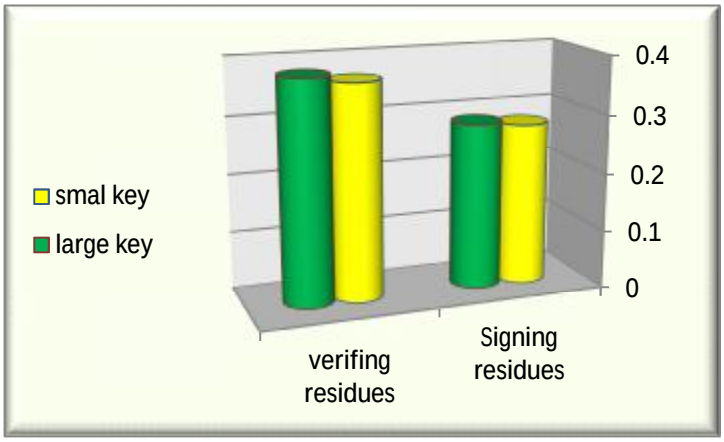


Figure 5.21: Signing and verifying of the residues

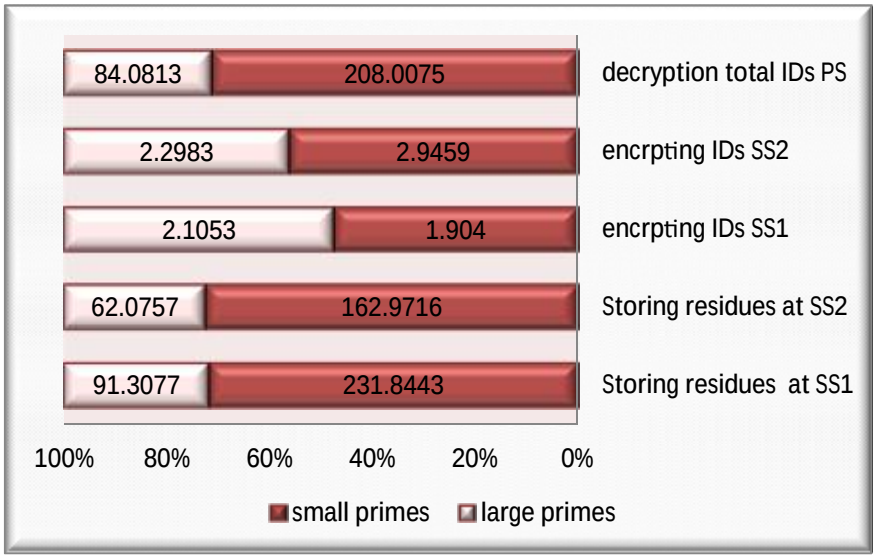


Figure 5.22: Storing, Encrypting and Decrypting of IDs between SS & PS

The time of the operations of creating, signing and encrypting FD at PS is compared with the time of decrypting verifying operations at RSs as shown in the Figure 5.23. Figure 5.24 shows the times taken by RS for encrypting the sent IDs to the SSs, and the time of decrypt its.

As mentioned previously, the information about the patients is taken from DB at PS and after the system completes all operations, the patient's data is stored in another DB at RS. The time of storing data is larger than the time for getting it.

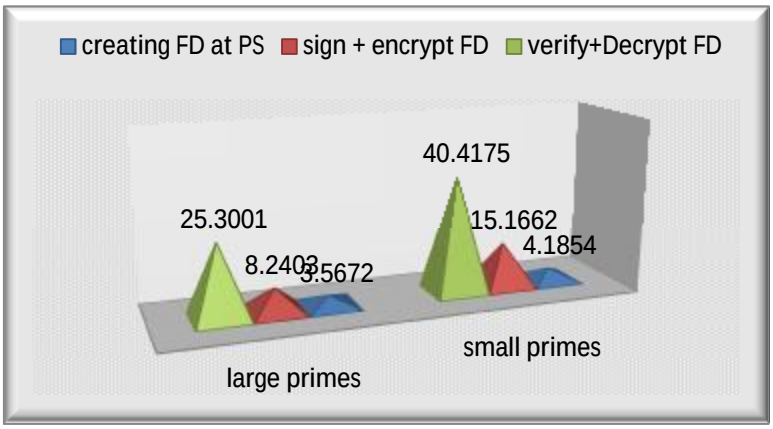


Figure 5.23: Creating, Signing, encrypting, verifying and decrypting FD

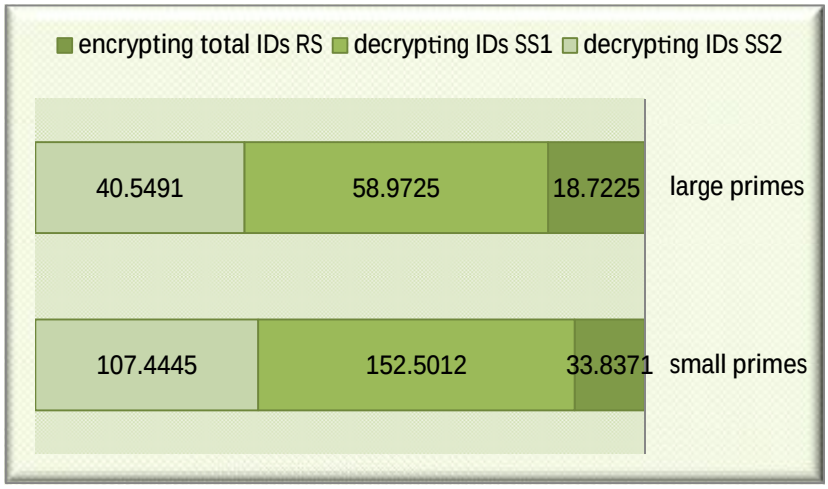


Figure 5.24: Encrypting and decrypting of IDs between RS & SS

The CC that is computed for the structure of the main operations in our system is illustrated in the Figure 5.25.

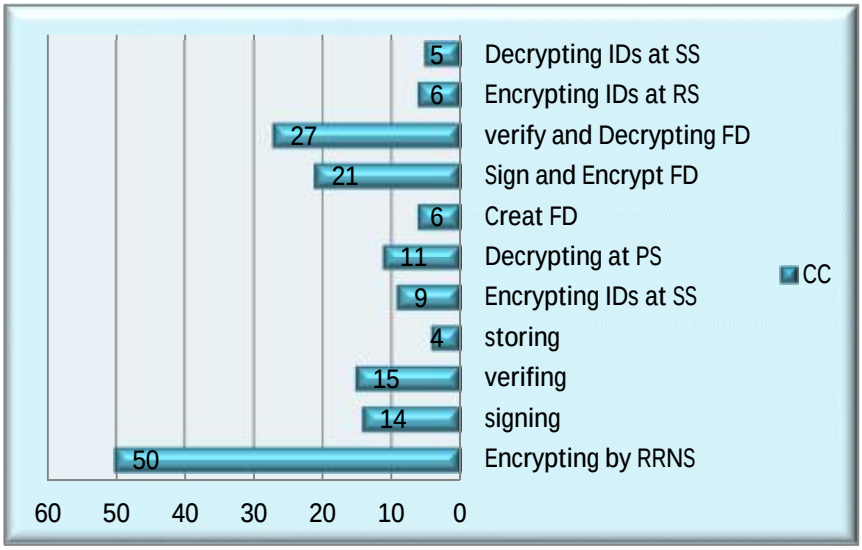


Figure 5.25: Cyclomatic Complexity of the system operations

Table 5.4 is abstracted the times of operations by using small and large moduli and also shown the Cyclomatic Complexity of these operations. The results shown that storing the residues in SS_s and decryption process by RSA at PS and SSs are taken large time, while the time of signing, verifying the residues, the encryption process at SSs and create FD is small time. The signing and encrypting FD at PS with verifying and decrypting FD at RS, and encrypting process at RS are taken intermediate time.

Table 5.4: Time and CC of all operation in the system

Operations	**Time of operation using Small primes**	**Time of operation using Large primes**	**CC**
Signing the residues	0.2794	0.286	14
Verifying the residues	0.3686	0.3804	15
Storing residues at SS_1	231.8443	91.3077	4
Storing residues at SS_2	162.9716	62.0757	4
Encrypting ID at SS_1	1.904	2.1053	9
Encrypting ID at SS_2	2.9459	2.2983	9
Decrypting ID at PS	208.0075	84.0813	11
Creating FD	4.1854	3.5672	6
Signing and encrypting FD	15.1662	8.2403	21
verifying and decrypting FD	40.4175	25.3001	27
Encrypting total IDs at RS	33.8371	18.7225	6
Decrypting IDs at SS1	152.5012	58.9725	5
Decrypting IDs at SS2	107.4445	40.5491	5
Encoding by RRNS			50

In hand of CC, encoding by RRNS, Signing and encrypting FD, and verifying and decrypting FD are executed with high CC. Signing and verifying the residues, Encrypting ID at SS_s, and Decrypting ID at PS have intermediate CC. Finally the following operation executed with less CC: storing residues at SS_s, Creating FD, Encrypting total IDs at RS, and Decrypting IDs at SSs.

Chapter six

Conclusion and Suggestions Future Work

Chapter six

Conclusion and Suggestions Future Work

6.1. Conclusions

Based on the results obtained from simulation and implementation of the system, some points have been concluded. These points are summarized below:

1. Using DSS at the top layer of WBAN architecture eliminates single point of failure problem because it distributes the storage of data on many servers spread on the network. DSS also helps in storing patient data over long periods of time with high reliability.
2. RRNS has used two libraries of moduli in the encoding process. After implementing the system with both libraries, it can be noted that using big prime integer moduli has increased the code efficiency, decreased encoding and decoding time, and reduced the size of the sent message between servers. This is especially useful in case of using public key cryptography.
3. Adding the signature on the residues, increases the security where any change on the sent residues can be detected at SSs before arriving to the RS.
4. The important part in the decoding process is FD. Therefore securing this part is essential point because without encrypting and signing FD, any unauthorized user can obtain the patient data. Therefore in the proposed system, encrypted and signed FDs are sent to the trusted RS.
5. In the implemented system, three algorithms have been used for decoding the RRNS. After analysis, it has been deduced that CRTI decodes the residues faster than classical CRT and BEX with MRC. But it has large complexity compared to the others, where CRT executes with the lowest complexity.

6. The dependability and confidentiality of the system have been ensured at first by using RRNS, where using the redundancy made the RS able to recover the residues up to $s \leq r$ erasure residues, and $\frac{r-s}{2}$ corrupt residues. Confidentiality is achieved when the RS cannot read any patient data unless getting the proper moduli.
7. Using RSA for encrypting the exchanged messages between servers such as storage locations from SSs to PS, FD from PS to RS, and storage locations from RS to SSs enhanced the confidentiality and authentication of messages. In these cases messages are protected from eavesdroppers and the identity of the sender is guaranteed where only authorized has access to the patient data.
8. It is obvious that our DSS can offer security services on more than one level and using several mechanisms. This significantly increases the difficulty of launching successful attacks on the system.

6.2. Suggestions for Future Work

There are some suggestions to develop our work in the future. The most promising of these are:

1. Implementing this system in the real life e-health application by using big DB and large scale network that is required to select a library of huge moduli numbers in order to achieve high security.
2. Using symmetric key cryptography instead of public key algorithms in case of using so much data in order to increase the security with less time. Then a comparison between the two approaches can be achieved.
3. Implementing the system with focusing on the Ad hoc routing aspects of the DSS.
4. Performing formal security protocol analysis for the interaction between various security mechanisms in the system. It is also possible to launch some types of attacks on the system and analyze their effects.
5. Enhancing New Chinese Remainder Theorem I (CRTI) by using Hardware (H/W) implementation.

Reference

Reference

[1] *Daniel Lachance and Glen E. Clarke, "CompTiA Security+ Certification Practice Exams Exam SYO-301", McGraw-Hill Osborne Media; 1 edition, pp.1-69, June 2011.*

[2] *Glenn Berg, "MCSE Training Guide: Networking Essentials", New Riders Publishing; 2nd edition, pp.1-660, November 1998.*

[3] *Sharam Hekmat, "communication networks", PragSoft Corporation, pp.1-198, 2005.*

[4] *Larry L. Peterson and Bruce S. Davie, "Computer Networks", Morgan Kaufmann; 3 edition, pp.4-838, May 2003.*

[5] *Benoît Latré, Bart Braem, Ingrid Moerman, Chris Blondia, and Piet Demeester, "A survey on wireless body area networks". Wireless Networks, Volume 17, pp. 1-18, January 2011.*

[6] *Erdal Cayirci and Chunming Rong, "Security in Wireless Ad Hoc and Sensor Networks", Wiley; 1 edition, pp.1-283, February 2009.*

[7] *Pravin Ghosekar, Girish Katkar,and Dr. Pradip Ghorpade, "Mobile Ad Hoc Networking: Imperatives and Challenges", IJCA Journal, pp.153-158, 2010.*

[8] *Jagannathan Sarangapani, "Wireless Ad Hoc and Sensor Networks Protocols, Performance, and Control", CRC Press; 1 edition, pp.1-538 April 2007.*

[9] *Saurabh Singh and Dr. Harsh Kumar Verma, "Security For Wireless Sensor Network", International Journal on Computer Science and Engineering (IJCSE), Volume 3, No.6, pp. 2393 – 2399, June 2011.*

[10] *John A. Stankovic, "Wireless Sensor Networks", University of Virginia, pp.1-20, June 2006.*

[11] *Ying Miao, "Wireless Self-Organization Networks APPLICATIONS OF SENSOR NETWORKS", Faculty of Engineering Sciences, pp.1-6, 2005.*

[12] *Jason Lester Hill, "System Architecture for Wireless Sensor Networks", University of California, Berkeley, pp.1 196, spring 2003.*

[13] *Hemanta Kumar Kalita1 and Avijit Kar, "WIRELESS SENSOR NETWORK SECURITY ANALYSIS", International Journal of Next-Generation Networks (IJNGN), Volume.1, No.1, pp.1-10, December 2009.*

[14] *I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey", International Journal of Computer and Telecommunications Networking, Volume 38, pp.393-422, March 2002.*

[15] *Sonia Waharte, Raouf Boutaba , Youssef Iraqi, and Brent Ishibashi, "Routing protocols in wireless mesh networks: challenges and design considerations", Springer Science + Business Media, Volume 29, pp.285-303, July 2006.*

[16] *Edoardo Amaldi, Antonio Capone, Matteo Cesana, and Federico Malucelli, "On the design of Wireless Mesh Networks", Department of Electronics and Information, pp.1-6.*

[17] *Ian F. Akyildiz, Xudong Wang, and Weilin Wang, "Wireless mesh networks: a survey", Computer Networks and ISDN Systems, Volume 47, pp.445 – 487, March 2005.*

[18] *Erik Karulf, "Body Area Networks (BAN)", a survey paper written under guidance of Prof. Raj Jain, pp.1-10, April 2008, eak2@cec.wustl.edu.*

[19] *Garth V. Crosby, Tirthankar Ghosh, Renita Murimi, and Craig A. Chin, "Wireless Body Area Networks for Healthcare: A Survey", International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC), Volume 3, No.3, pp.1-26, June 2012.*

[20] *Javed Ahmad and Fareeha Zafar, "Review of Body Area Network Technology & Wireless Medical Monitoring", International Journal of Information and Communication Technology Research, Volume.2, No. 2, pp.186 - 188, February 2012.*

[21] *Shahnaz Saleem, Sana Ullah, and Hyeong Seon Yoo,"On the Security Issues in Wireless Body Area Networks", International Journal of Digital Content Technology and its Applications, Volume.3, No.3, pp. 178-184, September 2009.*

[22] *Sofia Najwa Ramli M.Eng and Rabiah Ahmad PhD, "Surveying the Wireless Body Area Network in the realm of Wireless Communication", Information Assurance and Security (IAS), pp.58-61 December 2011.*

[23] *Yasmin Hovakeemian, Kshirasagar Naik, and Amiya Nayak, "A Survey on Dependability in Body Area Networks", pp. 10-14, March 2011.*

[24] *ICT Applications and Cybersecurity Division Policies and Strategies Department ITU Telecommunication Development Sector, "Implementing e-Health in Developing Countries Guidance and Principles", pp.1-53, September 2008.*

[25] *Meg Broderick and D. H. Smaltz, "E-Health Defined", May 2003*

[26] *Se Dong Min, Byoung Woo Lee, Sung Won Yoon, Young Bum Lee, Jin Kwon Kim, Myoungho Lee, and Cheol Oh Park, "Actual Condition of Korean e-Health: What Do Enterprisers Want for Developing e-Health Industry?", e-Health Networking, Application and Services, 2007 9th International Conference on, pp. 304 -307, June 2007.*

[27] *Nilmini S. Wickramasinghe, Adam M.A. Fadlalla, Elie Geisler, and Jonathan L. Schaffer, "A framework for assessing e-health preparedness", Int. J. of Electronic Healthcare, Volume.1, No.3, pp.316 – 334, 2005.*

[28] *Michael O. Rabin, "Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance", Journal of the Association for Computing Machinery, Volume 36, pp.335-348, April 1989.*

[29] *Chessa and P. Maestrini, "Dependable and Secure Data Storage and Retrieval in Mobile, Wireless Networks", International Conference on Dependable Systems and Networks, 2003. ", International Conference on Dependable Systems and Networks, 2003.*

[30] *Rudi Ball, James Grant, Jonathan So, Victoria Spurrett, and Rogério de Lemos, "Dependable and Secure Distributed Storage System for Ad Hoc Networks", Springer-Verlag Berlin, Heidelberg, pp.142-152, 2007.*

[31] *Qian Wang, Kui Ren, Wenjing Lou, and Yanchao Zhang, "Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance", INFOCOM 2009, IEEE, pp.954 – 962, April 2009.*

[32] *Y. S. Han, S. Omiwade, and R. Zheng, "Progressive Data Retrieval for Distributed Networked Storage", IEEE transactions on parallel and distributed systems, Volume.23, No.12, pp.2303- 2314, December 2012.*

[33] *Joonas Pääkkönen, "Distributed Storage for Proximity Based Services", School of Electrical Engineering, pp.1-48, 2012.*

[34] *Tanakorn Chareonvisal, "Implementing Distributed Storage System by Network Coding in Presence of Link Failure", School of Electrical Engineering Kungliga Tekniska Högskolan Stockholm, Sweden,pp.1-51, September 2012*

[35] *Maheswaran Sathiamoorthy, Megasthenis Asteris, Dimitris Papailiopoulos, Alexandros G. Dimakis, Ramkumar Vadali, Scott Chen, and Dhruba Borthakur, "XORing Elephants: Novel Erasure Codes for Big Data", PVLDB Volume.6, No.5, pp.325-336, 2013.*

[36] *Lluis Pamies-Juarez, Frédérique Oggier, and Anwitaman Datta, "Decentralized Erasure Coding for Efficient Data Archival in Distributed Storage Systems", Proceedings of the 14th International Conference on Distributed Computing and Networking (ICDCN), Volume 7730, pp.42-56, 2013.*

[37] *Jamil. Y. Khan and Mehmet R. Yuce ,"Wireless Body Area Network (WBAN) for Medical Applications", The University of Newcastle. Faculty of Science & Information Technology, School of Electrical Engineering and Computer Science , pp.31- 628, 2010.*

[38] *Sana Ullah, Pervez Khan, Niamat Ullah, Shahnaz Saleem, Henry Higgins, and Kyung Sup Kwak, "A Review of Wireless Body Area Networks for Medical Applications", International Journal of Communications, Network and System Sciences, pp.1-7, Aug 2010.*

[39] *chris otto, aleksandar milenković, corey sanders, emil jovanov, "system architecture of a wireless body area sensor network for ubiquitous health monitoring", Journal of Mobile Multimedia, Volume.1, No.4 pp. 307-326, 2006.*

[40] *Dave Singelée, Benoît Latré, Bart Braem , Michael Peeters , Marijke De, Peter De Cleyn , Bart Preneel , Ingrid Moerman, and Chris Blondia, "A Secure Low-Delay Protocol for Multi-hop Wireless Body Area Networks", Concerted Research Action (GOA) Ambiorics, pp.1-19, 2005.*

[41] *Julien Ryckaert, Claude Desset, Vincent de Heyn, Mustafa Badaroglu, Piet Wambacq, Geert Van der Plas, and Bart Van Poucke, "Ultra-WideBand Transmitter for Wireless Body Area Networks". http://www.eurasip.org/Proceedings/Ext/IST05/papers/266.pdf*

[42] *Hadda Ben Elhadj, Lamia Chaari, and Lotfi Kamoun, "A Survey of Routing Protocols in Wireless Body Area Networks for Healthcare Applications", International Journal of E-Health and Medical Communications, Volume.3, pp.1-18, April-June 2012.*

[43] *Benoit Latre, "Betrouwbare en energie-efficiënte netwerkprotocollen voor draadloze Body Area Networks - Reliable and Energy Efficient Network Protocols for Wireless Body Area Networks", Ghent University Faculty of Engineering Department of Information, pp.1-258, June 2008.*

[44] *Kyung Sup Kwak, Sana Ullah, and Niamat Ullah, "An Overview of IEEE 802.15.6 Standard", International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL) in Rome, Italy, pp.1-6, Feb 2011.*

[45] *Eric Lam, "Performance Characterization of 802.15.6 Wireless Body Area Network (WBAN) MAC Protocol", 2012, ericflam@stanford.edu*

[46] *Steve He, Eric Wang, "AN OVERVIEW OF WBANS", http://web.cs.wpi.edu/~rek/Adv_Nets/Fall2011/WBANs.pdf*

[47] *Daisuke Takahashi, Yang Xiao, and Fei Hu," LTRT: Least Total-Route Temperature Routing for Embedded Biomedical Sensor Networks Global Telecommunications Conference, GLOBECOM . IEEE -GLOBECOM , pp. 641-645, 2007.*

[48] *FrederiqueOggier and Anwitaman Datta, "Self-repairing Homomorphic Codes for Distributed Storage Systems", IEEE International Conference on Computer Communications, pp.1-10, Jul 2010.*

[49] *MountainView ITSM, "Centralized vs. Distributed Computing", White Paper Discussion, October 29, 2007.*

[50] *Oliver Schneider, "Trust Aware Social Networking: A Distributed Storage System based on Social Trust and Geographical Proximity", Fachbereich für Informatik der Freien, University of Berlin, pp.1-72, January 2009.*

[51] *Sameer Pawar, Salim El Rouayheb, and Kannan Ramchandran, "Securing Dynamic Distributed Storage Systems against Eavesdropping and Adversarial Attacks", IEEE Transactions on Information Theory, Volume 57, pp.6734-6753, 2011.*

[52] *Salim El Rouayheb and Kannan Ramchandran, "Fractional Repetition Codes for Repair in Distributed Storage Systems", CoRR, Annual Allerton Conference on Communication, Control, and Computing - Allerton, September, pp.1-8, 2010.*

[53] *Alexandros G. Dimakis, P. Brighten Godfrey, Martin J. Wainwright, and Kannan Ramchandran, "The Benefits of Network Coding for Peer-to-Peer Storage Systems", Third Workshop on Network Coding, Theory, and Applications (NETCOD), pp.1-6, January 2007.*

[54] *Rekha Bachwaniy, Leszek Gryzz, Ricardo Bianchiniy, and Cezary Dubnicki, "Dynamically Quantifying and Improving the Reliability of Distributed Storage Systems", Reliable Distributed Systems, 2008. SRDS '08. IEEE Symposium on, pp.85-94, October 2008.*

[55] *Alexandros G. Dimakis, P. Brighten Godfrey, Yunnan Wu, Martin J. Wainwright, and Kannan Ramchandran, "Network Coding for Distributed Storage Systems", IEEE Transactions on Information Theory, Volume.56, No.9, pp.4539-4551, September 2010.*

[56] *MARTIN PLACEK and RAJKUMAR BUYYA, "A Taxonomy of Distributed Storage Systems", The University of Melbourne Australia, pp.1-53, July 2006.*

[57] *Vishal Kher and Yongdae Kim, "Securing Distributed Storage: Challenges, Techniques, and Systems ACM International Workshop on Storage Security and Survivability, pp.9-25, 2005.*

[58] *Ming Li, Wenjing Lou, and Kui Ren, "data security and privacy in wireless body area networks", IEEE Wireless Communications, Volume 17, pp.51-58, February 2010.*

[59] *Mohammed Raza Kanjee, Kalyani Divi, and Hong Liu, "A Two-Tiered Authentication and Encryption Scheme in Secure Healthcare Sensor Networks", International Conference on Information Assurance and Security, pp.271-276, Aug 2010.*

[60] *William Stallings, "Cryptography and Network Security Principles and Practices", Fifth Edition, Prentice-Hall, 2011, pp.1-900, November, 2005.*

[61] *Jingwei Liu and Kyung Sup Kwak, "Hybrid Security Mechanisms for Wireless Body Area Networks", Ubiquitous and Future Networks (ICUFN), 2010 Second International Conference on, pp.98-103, June 2010.*

[62] *Wolfgang Leister, Habtamu Abie, and Ilangko Balasingham, "Threat Assessment of Wireless Patient Monitoring Systems", Information and Communication Technologies: From Theory to Applications, 2008. 3rd International Conference on, pp.1-6, April 2008.*

[63] *Chol-soon Jang, Deok-Gyu Lee, and Jong-wook Han, "A Proposal of Security Framework for Wireless Body Area Network", International Conference on Security Technology, pp.202-205, Dec. 2008.*

[64] *Shervin Amini, Richard Verhoeven, Johan Lukkien, and Shudong Chen," Toward a Security Model for a Body Sensor Platform", International Conference on Consumer Electronics (ICCE), pp.143-144, Jan 2011.*

[65] *Lin Yao, Bing Liu, Kai Yao, Guowei Wu, and Jia Wang, "An ECG-Based Signal Key Establishment Protocol in Body Area Networ", Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing, pp.233-238, October 2010.*

[66] *Lie-Liang Yang and Lajos Hanzo, "Coding Theory and Performance Of Redundant Residue Number System Codes", IEEE Transactions on Information Theory - TIT, pp.1-40, Aug. 1999.*

[67] *AVIK SENGUPTA, "REDUNDANT RESIDUE NUMBER SYSTEM BASED SPACE-TIME BLOCK CODES", KANSAS STATE UNIVERSITY, Department of Electrical and Computer Engineering Manhattan, Kansas, August 2012.*

[68] *S. Chessa, R. Di Pietro, and P. Maestrini, "Dependable and Secure Data Storage in Wireless Ad Hoc Networks: An Assessment of DS^2", International Federation for Information Processing, pp.184 –198, 2004.*

[69] *Jean-Claude Bajard and Thomas Plantard, "RNS bases and conversions", Society of Photo-Optical Instrumentation Engineers, Bellingham, WA, INTERNATIONAL, Volume. 5559, pp. 60-69, 2004.*

[70] *Avi Kak," Computer and Network Security ", Avinash Kak, Purdue University, April 24, 2010.*

[71] *Amusa, K. A. and Nwoye E. O., "NOVEL ALGORITHM FOR DECODING REDUNDANT RESIDUE NUMBER SYSTEMS (RRNS) CODES", IJRRAS, Volumes 12, July 2012.*

[72] *Narendran Narayanaswamy, "OPTIMIZATION OF NEW CHINESE REMAINDER THEOREMS USING SPECIAL MODULI SETS", Louisiana State University and Agricultural and Mechanical College, pp.1-68, December 2010.*

[73] *Evgeny Milanov, "The RSA Algorithm", University of Washington, June 2009.*

[74] *Manuel Mogollon, "Cryptography and Security Services: Mechanisms and Applications", CyberTech Publishing, pp.1-489, 2007.*

[75] *Charles H. Romine, "Secure Hash Standard (SHS)", FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, pp.1-35, March 2012.*

Appendix

Appendix A
Network Setting

A.1: Creating local network

The steps of creating local network at PS are shown in the Figure A.1. PS after being connected to the built network, it will wait other servers to be connected with it. The other servers communicate with PS by entering the right password.

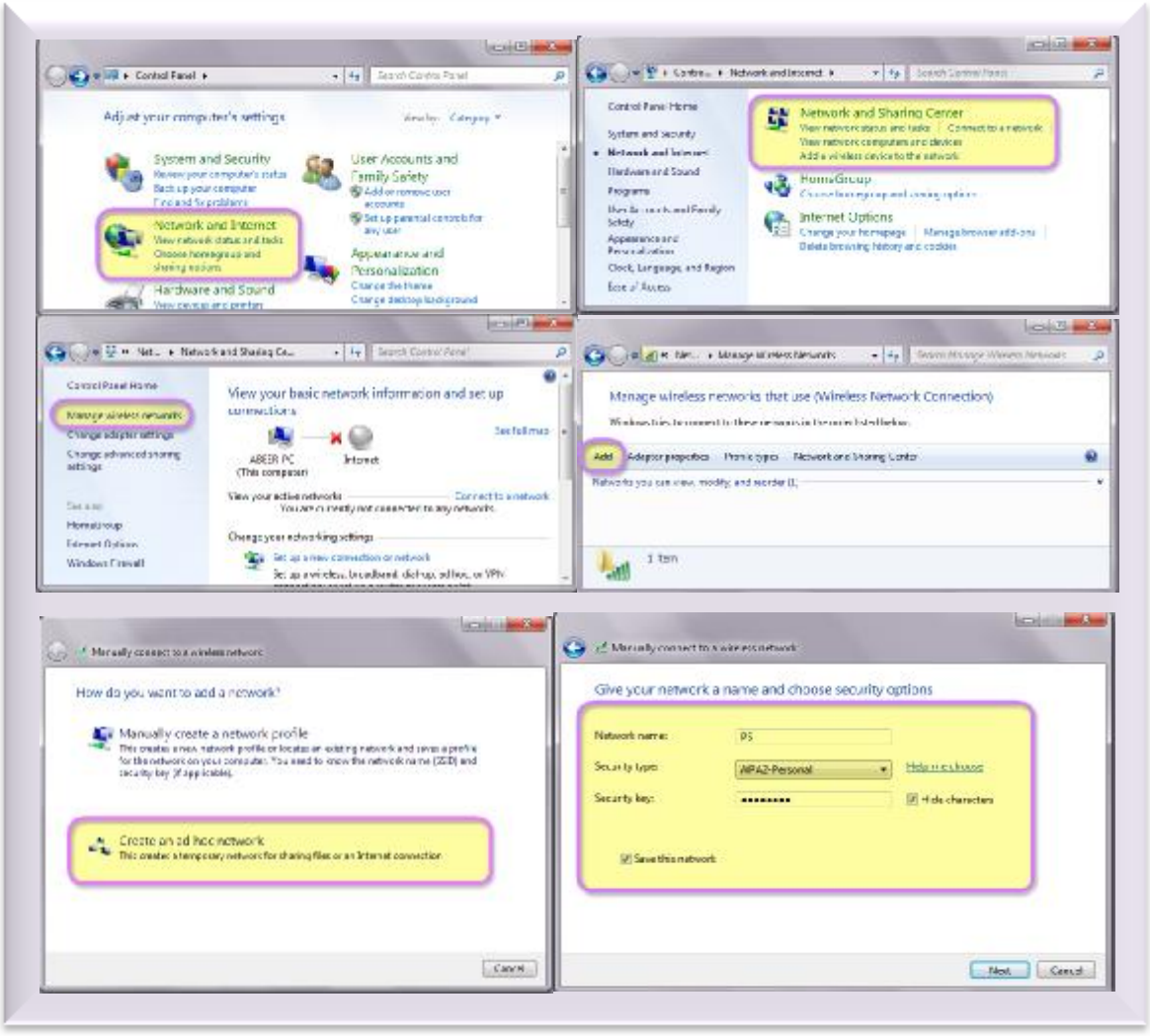


Figure A.1: Network Creation

A.2: Setting Internet Protocol (IP)

Each server as mentioned has specified IP address. The processes of setting IP address are shown in the Figure A.2. These processes must be applied in every server. The IP for each server in our developed system prototype are explained in the Table A.1.

Table A.1: IPs address and subnet mask for each Server

Server name	IP address	Subnet mask
PS	192.168.1.3	255.255.255.0
SS1	192.168.1.4	255.255.255.0
SS2	192.168.1.5	255.255.255.0
RS1	192.168.1.10	255.255.255.0

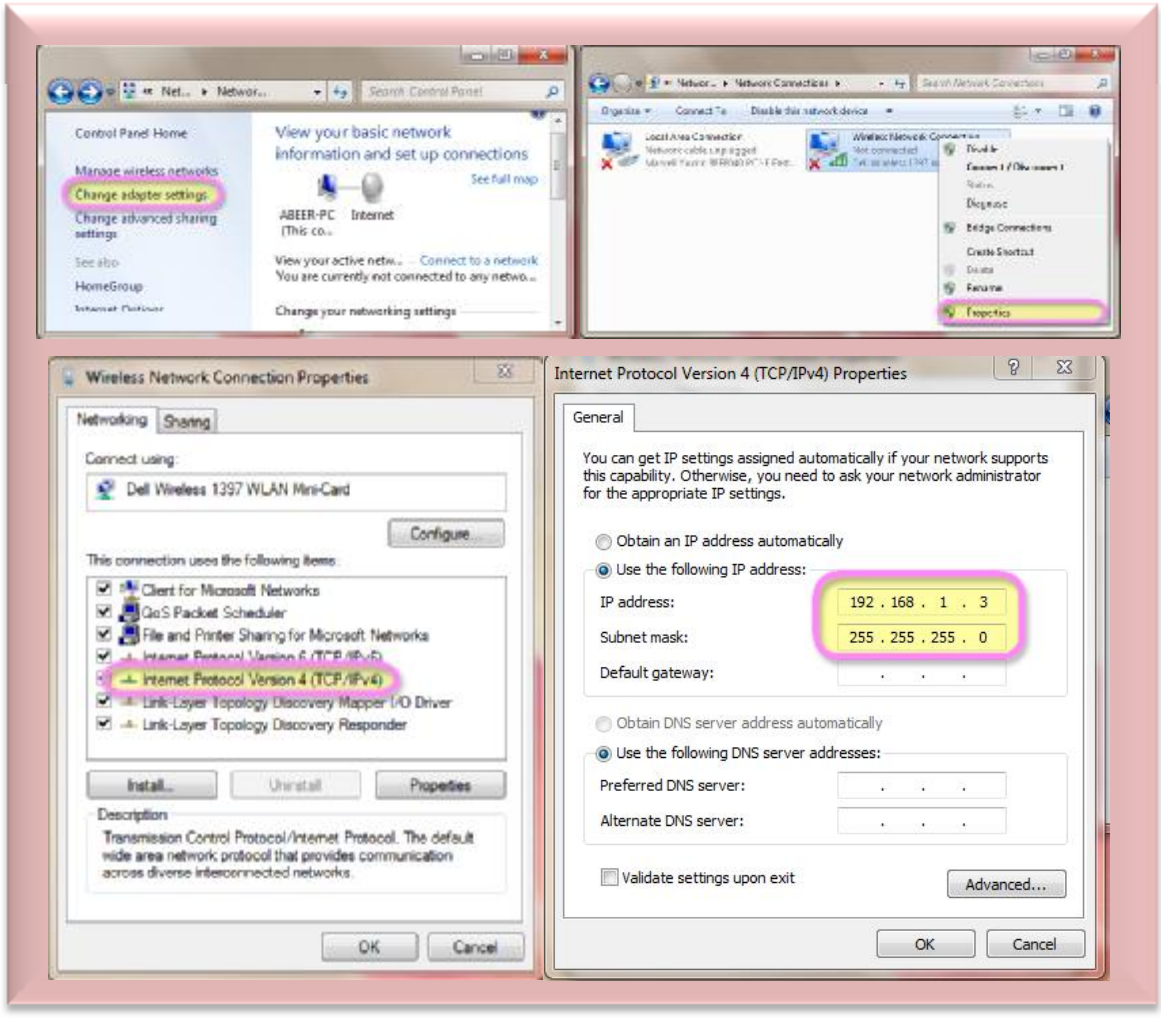


Figure A.2: Setting IP address

الخلاصة

شبكات منطقة الجسم اللاسلكية (WBAN) تقنية حديثة أُستخدمت في مجال الرعاية الصحية ، تتألف من عدد من المتحسسات الذكية التي تجمع البيانات الطبية من جسم الانسان وتنقلها بواسطة الشبكات اللاسلكية المحلية (WLAN) الى جهاز شخصي الذي بدوره ينقل بيانات المريض الى خادم طبي عبر الانترنت، حيث ان الخبراء يقومون بتحليل هذه البيانات التي تكون حساسة وحرجة، لهذا أمنية وخصوصية هذه البيانات يجب أن تكون عالية. وإنَّ تعرض هذه البيانات للتلاعب سواءالسرقة أو التغير يهدد خصوصية المريض.

في هذه الرسالة، تم اقتراح نظام الخزن الموزع (DSS) الذي يعتمد على توزيع خزن بيانات المريض بين عدة خوادم موثوقة ومنتشرة في الشبكة بدلا من تخزينها في خادم مركزي واحد. نظام الخزن الموزع يستند على شفرات المحو لإنجاز هذا الغرض حيث إنه يتألف من خادم أساسي (PS) واحد الذي يحتوي على قاعدة البيانات الرئيسية، إثنان من الخوادم الخازنة (SSs) ألمسؤولة عن تخزين بيانات المريض، وخادم قراءة واحد (RS) الذي يقوم بتجميع معلومات المريض من الخوادم الخازنة لقراءتها. PS يشفر البيانات بأستخدام تقنية (RRNS) التي تعتمد على مكتبة من المفاتيح لتوليد البقايا. ثلاث خوارزميات طُبِقت لفك تشفير RRNS وهي: CRT, ,BEX with MRC و CRT I. كما إن خوارزمية RSA أستخدمت لتشفير المعلومات المتبادله بين الخوادم، تم توقيع البقايا و واصف الفايل FD باستخدام DSA قبل البدء بعملية الارسال.

بعد تنفيذ النظام بواسطة التمثيل الحاسوبي، تبين انه كلما كانت المفاتيح كبيره، كان وقت التشفير وفك التسفير وطول الرسالة المرسلة اقل و كانت كفاءة الشفرة عالية. خوارزمية (CRT I) تعتبر الافضل لفك تشفير RRNS لانها تجعل الحسابات سريعة وفعالة مع اقل كلفة. أن استخدام تقنية RRNS جهزت السرية والاعتمادية لبيانات المريض، كما إنَّ إستخدام DSA ساعد في التحقق من سلامة البيانات المرسلة وإنَّ استخدام خوارزمية RSA حقق السرية والمصداقية وعدم النكران.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة الانبار
كلية الحاسوب - قسم علوم الحاسبات

تحسين خوارزميات المفتاح المعلن الموزعة لنظام الصحة الالكترونية

رسالة مقدمة الى

قسم علوم الحاسبات – كلية الحاسوب – جامعة الانبار وهي جزء من

متطلبات نيل درجة الماجستير في علوم الحاسبات

قدمت من قبل

عبير داود سلمان النعيمي

بإشراف

أ.د. سفيان تايه فرج الجنابي

و

د.علي جبير داود

١٤٣٥ هـ

٢٠١٤م