SMARTPHONE APPLICATION ARCHITECTURE AND SECURITY FOR PATIENT VITAL SIGNS SENSORS AND INDICATORS

by

Orika Orrie

Submitted in partial fulfilment of the requirements for the degree
Master of Engineering (Computer Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

July 2016

SUMMARY

SMARTPHONE APPLICATION ARCHITECTURE AND SECURITY FOR PATIENT VITAL SIGNS SENSORS AND INDICATORS

by

Orika Orrie

Supervisor: Dr GP Hancke

Department: Electrical, Electronic and Computer Engineering

University: University of Pretoria

Degree: Master of Engineering (Computer Engineering)

Keywords: Medical data, patient, security, encryption, data transfer, speed

South Africa is a developing country with great potential to be leaders in technology and research, especially in the medical field. Rural areas in many countries do not have access to basic healthcare services due to the distance and inaccessibility of these services. Currently people living in the rural areas in South Africa are required to rely on the people within the area, who may not be trained; on doctors who make house calls, who may not be able to access the patient in time or on finding transport to the nearest hospital, which may be hundreds of kilometres away. This leads to many rural residents not seeking aid for aliments thereby often lowering life expectancy.
South Africa has many world-renowned medical practitioners who would be able to assist the residents in these areas if there were methods for observation and recording of health statuses without the need for either party to travel.

This dissertation studied and developed a method to assist not only the residents in rural areas, but also urban residents to record their vital signs without the assistance of a licenced medical practitioner, to upload the data to a database and to then allow the data to be viewable by the medical practitioner who may be situated elsewhere in South Africa or the world. This system allows for the elimination of human error when recording vital sign data as recording is not done through human intervention. Through the use of communications technologies such as Bluetooth, NFC and Wi-Fi a system was designed which ensures that a patient can record medical data without the presence of a medical

practitioner, the patient can access previous health records and readings and the patient can give a new medical practitioner a full medical history. The patient's data has been secured using AES and RSA encryption as well as verification through hash values at all points of transfer and access is granted to the patients' medical data only through the patient or a licenced medical practitioner. The data recording and transfer has been completed taking into consideration all the medical legislation and laws in South Africa.

This system allows the South African medical health sector to service all South Africa residents, including the residents in rural areas.

OPSOMMING

SLIMFOONTOEPASSING-ARGITEKTUUR EN -SEKURITEIT VIR PASIËNTLEWENSTEKEN-SENSORS EN -AANWYSERS

deur

Orika Orrie

Studieleier: Dr GP Hancke

Departement: Elektriese, Elektroniese en Rekenaar-Ingenieurswese

Universiteit: Universiteit van Pretoria

Graad: Magister in Ingenieurswese (Rekenaar-Ingenieurswese)

Sleutelwoorde: Mediese data, pasiënt, sekuriteit, enkripsie, data-oordrag, data-spoed

Suid-Afrika is 'n ontwikkelende land met geweldige potensiaal om die leiding te neem ten opsigte van tegnologie en navorsing, veral in die mediese veld. Landelike gebiede in verskeie lande het nie toegang tot basiese gesondheidsorg nie as gevolg van afstand en ontoeganklikheid van die nodige dienste. Tans is mense woonagtig in die landelike gebiede van Suid-Afrika afhanklik van die bekikbare dienste in hul onmiddellike omtrek, wat dikwels impliseer dat die nodige opleiding ontbreek. Dokters wat huisbesoek aflê is dikwels nie daartoe instaat is om die pasiënt betyds te bereik nie; of om vervoerreëlings te tref na die naaste hospitaal wat honderde kilometers vêr mag wees. Dit het tot gevolg dat baie landelike inwoners geen toegang tot mediese dienste het nie en word lewensverwagting verlaag. Suid-Afrika beskik oor werêldbekende mediese praktisyns wat daartoe instaat sou wees om die inwoners sinvol te ondersteun mits daar die moontlikheid geskep kan word dat dit kan plaasvind sonder dat enige van die partye hoef te reis.

Hierdie studie het 'n metode nagevors en ontwikkel wat die opname van lewenstekens sonder die bystand van 'n gekwalifiseerde geneesheer, nie alleenlik vir mense in landelike gebiede nie maar ook in stedelike gebiede, moontlik maak. Die metode bied die moontlikheid om die data na 'n databasis te versend en bemagtig dan die geneesheer om die data te besigtig vanaf enige verafgeleë plek in Suid-Afrika of die wêreld. Hierdie stelsel elimineer grotendeels menslike foute tydens die opname van 'n lewensteken aangesien daar nie 'n mens is wat die waarneming behartig nie. 'n Stelsel is ontwikkel wat toelaat dat 'n pasiënt mediese data kan opneem sonder 'n geneesheer en toegang verkry tot

historiese mediese opnames en lesings deur die gebruik van kommunikasie tegnologieë soos Bluetooth, NFC en Wi-Fi. 'n Pasiënt kan ook 'n volle mediese geskiedenis oorhandig aan 'n nuwe geneesheer. Die pasiënt se data word beskerm deur middel van AES en RSA enkripsie sowel as verifikasie deur ‘n hutssleutel tydens data-oordrag. Toegang tot die data word alleenlik gegee deur die pasiënt of 'n gelisensiëerde geneesheer. Die dataopname en -oordrag is op so 'n manier opgestel om aan die nodige mediese wetgewing in Suid-Afrika te voldoen.

Hierdie stelsel verskaf die moontlikheid aan die Suid-Afrikaanse mediese bedryf om ‘n diens te lewer aan alle Suid-Afrikaanse inwoners, insluitend diegene in landelike gebiede.

LIST OF ABBREVIATIONS

AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
DES	Data Encryption Standard
ECG	Electrocardiogram
HMAC	Hash Message Authentication Code
ID	Identity
IP	Internet Protocol
MD5	Message Digest Algorithm
MFA	Multi-factor Authentication
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
RFID	Radio Frequency Identification
RTD	Record Type Definition
RX	Receive
SHA	Secure Hash Algorithm
TEE	Trusted Execution Environment
TNF	Type Name Format
TX	Transmit
UART	Universal Asynchronous receiver/transmitter
URI	Uniform Resource Identifier
USB	Universal Serial Bus
WPAN	Wireless Personal Area Network

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 PROBLEM STATEMENT

1.1.1 Context of the problem

The digital management and storage of data has become widespread over multiple fields of study. Within South Africa the patient readings derived from multiple medical devices and the notes of the medical practitioner are recorded in a document and stored in a dossier [1]. This patient information can be specific to the hospital, the doctor or the medical facility and is generally not shared among facilities unless there is a great need for previous information. Once the documents are requested at a different facility a personnel member in the original facility will need to retrieve the data and email or fax the document to the medical practitioner in need of it. In an emergency this process can be time consuming.

Within South Africa there are multiple rural areas which are inaccessible by road; these areas cause difficulties when a medical practitioner needs to monitor a patient's vital signs over long periods [2]. In addition, certain urban-dwelling patients have difficulty recording accurate vital signs which need to be monitored constantly and consistently. There is a need for a method of vital sign recording where the patient will not need to be moved to a more urban area and the medical practitioner will not need to constantly travel to the rural area, as well as taking away the responsibility of recording the patient's data by hand and rather entrusting it to a digital recording medium.

A system is needed which optimises these needs through the use of the internet and computing technology.

1.1.2 Research gap

The dissertation researched the current need for digital management and storage of medical data within South Africa. There are currently no protocols created using a South African context which allows for the manipulation and storage of sensitive medical data which includes the South African governmental legislation and the Personal Protection of Information Act (POPI).

1.2 RESEARCH OBJECTIVE AND QUESTIONS

The main objective for this dissertation was to identify the gaps in the medical field regarding the legal transfer of patient medical data. This will be completed through the design of a security protocol encapsulating all the medical legislative requirements and the technical requirements.

To address the objective of this dissertation, the following questions have been posed:

1. Which security services and processes are required to satisfy legal requirements with regard to medical information collected by autonomous medical sensing systems?
2. How can these services and processes best be implemented on easily available consumer electronic devices, which would lower the cost of these systems in public healthcare?
3. Is it feasible for these services to be implemented in such a way that critical data is collected in a reliable and timely manner?
4. What performance can be achieved in terms of reliability and latency of sensed data if both the legal aspects and the platforms' requirements and limitations are taken into account?

1.3 APPROACH

This dissertation outlines the design and the performance of a security protocol for the legal collection of patient medical data using a mobile application. Protocols need to allow for timely transfer of data between a medical device and the patient's device, patient's device and medical database, and also between the patient's security token (NFC ID card) and the device. The hypothesis is to demonstrate that such a system, adhering to current legislation, can be implemented using protocols optimised to the mobile platform and secondary devices used (medical devices and security tokens).

The approach taken in this dissertation covers a review of documentation and literature surrounding wireless data transmission in hospital environments globally and nationally,

the security issues surrounding this situation and the management of multiple sensor devices as part of the literature study in Chapter 2.

In Chapter 3 the architecture and implementation of the patient-based monitoring system is covered with four main focuses; the proposed system configuration, the transfer of medical information between the device and the smart phone application, the NFC token based authentication of the data transmitted to the database and the optimal cryptographic design for the data transfer and throughput.

Chapter 4 then covers the testing procedure for the architecture implementations described in Chapter 3 and the results found.

Chapter 5 is a discussion on the adherence of the system to current medical health record legislation and Chapter 6 documents any conclusions which can be drawn from the dissertation.

The steps in the scientific process are followed for this dissertation. Listed in order the process is: problem identification, literature review, formulate hypothesis, design experiments, collect and compare data, conclusion. During problem identification the legislative requirements are mapped to the necessary technical requirements, followed by a review of possible protocols and cryptographic methods for realising these requirements. A test system will then be implemented to evaluate the performance of the protocols in a realistic setting.

1.4 RESEARCH GOALS

The goal of the research completed is to provide a working mobile platform for medical use. The platform should adhere to all medical legislation with regard to the recording and tracking of medical health data collected by the medical application. In addition, the medical application should perform in accordance with all security concerns and speed of transmission concerns for the transfer and recording of the medical health records by both the medical practitioner and the patient.

1.5 RESEARCH CONTRIBUTION

This dissertation contributed protocols needed in smart phone applications when working with sensitive medical data subject to patient consent. The dissertation focused specifically

on system design and evaluation that was suited to South African public hospitals and legislation. In addition, the architecture and design for the mobile monitoring of the performance (reliability and latency) of different cryptographic functions to implement the required approach was evaluated on a smartphone application.

1.6 OVERVIEW OF STUDY

The study focuses on the design of a complete data collection system for patient vital signs, personal information and past data collected. The system covers the data collection from the medical devices to the smartphone device; the authorisation and access needed to obtain the patient medical and personal information; and the method of storage of the data on a global database and website. The study covers all the legal and legislative aspects which can come up when considering confidential patient medical data.

CHAPTER 2 LITERATURE STUDY

2.1 CHAPTER OBJECTIVES

This chapter poses a review of the documentation and literature surrounding wireless data transmission in hospital environments, the security issues surrounding this situation and the management of multiple sensor devices. This chapter intends to assist in the decision on which wireless technology to use in the mobile health sensor application. Bluetooth and Near-Field Communication (NFC) are to be investigated as these technologies are widely used in various other studies [11], [12], [23], while a range of data security and management techniques was used.

There is a need in South Africa for widely available medical devices to digitally manage medical data from patients [1]. An off-the-shelf device such as a smart phone or tablet can interface with wireless medical devices and be used to automate medical data collection. However, medical records are sensitive with regard to data integrity and confidentiality while mobile devices are often seen as less secure than devices which are made for the purpose of keeping medical records safe. The device should be a means to accurately store medical records for use by various doctors and medical health practitioners. The research investigates the feasibility of using current mobile devices to securely manage medical information in addition to developing and evaluating suitable cryptographic protocols to support this capability. The research takes into consideration Trusted Execution Environment (TEE) developments and NFC in proposing protocols for linking practitioner and patient ID to data, to manage keys used to communicate with medical devices, and to secure data for storage on back-end systems.

2.2 WIRELESS COMMUNICATION PROTOCOLS

The design of the smartphone application has many components. One of the main components is the wireless communication and data transfer protocols used. NFC and RFID tags are used to store the critical and sensitive patient data, therefore the transfer method in addition to the storage method needed to be secure. Currently there are medical

devices available which communicate via wireless mediums such as Bluetooth and Wi-Fi [2], [3], [11], [12], and [23].

Wi-Fi, ZigBee, Bluetooth and NFC are the technologies chosen for this literature study as these technologies are the most widely implemented [12]. The advantages and disadvantages will be evaluated and a decision on the most reliable and secure protocol will be made for the purposes of this dissertation.

CodeBlue [14] is an emergency response system which operates using ad hoc infrastructure. CodeBlue reduced the amount of packet loss which can be found in many systems of a similar nature. The system operates by sending streams of vital signs to a central device; these streams are sent directly from the sensing nodes.

CodeBlue was set up for a major hospital, the ad hoc protocol was chosen for the benefit of the signal's ability to extend across a major building or a group of adjacent buildings. The data transmission was secured using error detecting and correcting codes.

The challenges encountered by the CodeBlue system were how to make the communication reliable as well as secure. The system also needed to prioritise critical patient data over normal data. A spanning-tree multi-hop routing algorithm was developed to minimise the packet loss in the system. The spanning-tree multi-hop algorithm was coded to be used with constant data streams; the data was then sent to a database which monitors patients' vital signs in real-time. Although this system has the advantage of real time patient monitoring; errors come up in the security functions as the data needs to be transmitted fast as well as securely.

Data for a patient health system needs to be stored within the database when the information is available. Functionality was tested for intermittent sending of data as opposed to a constant stream of data as the data costs for constant live data may be too high or impossible for rural areas. Wi-Fi was determined as the most reliable method for hospital data transfer [3] as communication interference does not affect the transmission once the router has been correctly configured. Access privileges to the data needed to be multi-tiered [3] to control access to the patient's private data. Access needs to be granted only to doctors in contact with the specific patient and to certain medical staff involved with the patient; this access is removed once the patient is discharged, ends the consultation or revokes permissions.

Radio frequency identification or RFID is an emerging technology which was created with the intention of replacing barcodes on products in many ways [4], [21]. An RFID reader activates a transponder in the RFID tag which remotely reads the data stored on the tag or writes data to the tag. The Taipei Medical University Hospital was one of the first hospitals to consider RFID in the medical field [4]. The university used the tags to track patient locations around the building; if the patient came into close contact with infected or contagious patients a warning bell sounded and an emergency exposure protocol was followed to contain the situation.

To secure the data the Taipei Medical University Hospital and later the Wan Fang Hospital in Taipei implemented two solutions for testing. The first solution was to secure the data using public and private key encryption on each tag, the system consisted of various levels and only the smartphone with the correct level of authority gained access to the relevant data. The University ran into a problem once the data grew in size as the NFC tags did not have enough storage space for the encrypted patient data.

The second iteration implemented was to store only the linking data on the patient's tag, i.e. the data which linked the patient to the database and to the doctor or nurse in charge of the patient. Unfortunately, the administration involved in storing the data correctly and linking the data correctly to existing and incoming patients was huge on the part of the server. Additional security measures would also need to be implemented to secure the data on the server side and on the smart device side [17], [19]. Each smart device or smartphone needed the encryption keys as well as a privilege level assigned to the staff members [20].

The IEEE 802.15.1 standard, Bluetooth, is based on a wireless radio system. It has been designed to work over shorter distances and built as a cheap device to replace cables [6]. In Bluetooth communications there are 2 major connectivity protocols, a piconet and a scatternet. A piconet is a Wireless Personal Area Network (WPAN) formed when a master a Bluetooth device serves as a master with one or more other Bluetooth devices serving as slaves. A scatternet is formed when two or more piconets are connected at one point.

In one case an electrocardiogram (ECG) monitor transmitted data using Bluetooth and detected if any abnormalities occurred in the ECG recorded, the data was then sent to a doctor's smartphone and notified the doctor of the severity [12]. The challenges discovered in the study were the errors occurring during the data transmission process, however when

tested using a ZigBee device, the data transmitted via Bluetooth was more reliable. In Table 1 the results of the tests performed in this study are shown. Bluetooth is currently the most reliable form of data transmission, the reliability and efficiency of the programming is much higher than that of a ZigBee device.

Table 1: Typical comparison parameters of the wireless protocols [12]

Standard *IEEE Spec.*	*Bluetooth* *802.15.1*	*ZigBee* *802.15.4*	*Wi-Fi* *802.11a/b/g*
Max data rate (>bit/s)	0.72	0.25	54
Bit time (µs)	1.39	4	0.0185
Max data payload (bytes)	339 (DH5)	102	2312
Max Overhead (bytes)	158/8	31	58
Coding efficiency* (%)	94.41	76.52	97.18
*Unapproved 802.15.3a			

According to the studies listed here, the most popular choices for wireless data transmission are Bluetooth and NFC.

2.3 SECURITY OF THE MEDICAL DATA

There are multiple threats that are faced when storing data within a wireless sensor network [2], [4], [6], [13], [14], [17], [18], [24]. The sensors used are usually not protected against illegal data access and are subjected to data compromise. For example, if a patient's results are recorded and then encrypted and stored to the sensor node with the encryption key, a person with the correct tools can then retrieve the data.

The local servers where the data is stored may not be trustworthy as persons with malicious intent may try to access the system to obtain patients' private information.

Another form of access to the server is when a node fails and it is not detected by the system. If a person who has malicious intent discovers this, the node may be replaced with

a fake sensor in order to masquerade authentic nodes. The attacker could also replace authentic nodes with fake ones to allow the information to be diverted to a system which the attacker has access to.

2.3.1 Trusted Execution Environment

A secure environment needs to be built in order to protect the data. A Trusted Execution Environment (TEE) is one such option [22]. A TEE is a secure section of the application processor in an electronic device. This can be seen as a secure facility with various safeguards; these safeguards need to be passed in order to gain access to the information within.

To prevent a compromise of the system the TEE contains certain security requirements which are the basis for the protection [13], [22].

1. Confidentiality: The patient's data needs to be kept confidential during the storage life.
2. Dynamic integrity assurance: Patient data cannot be illegally modified while in storage.
3. Dependability: The data must be available even in the case of a node failure or system compromise.
4. Access control: Only authorised access should be allowed to the data such as by relevant doctors and nurses.
5. Accountability: There should be a method to track the usage and any unauthorised access through authorised means should be detected and the person responsible should be identified.
6. Revocability: If a user is detected to be using the system maliciously, it should be possible for the user to be removed from the system.
7. Non-repudiation: The nodes should be able to identify and verify the data retrieved.
8. Authentication: The sender and node should have accurate authentication before the data is stored.
9. Availability: the data should be accessible even under denial of service attacks.

2.3.2 Data integrity and confidentiality

When designing the system, keeping data confidential is a top priority. This can be achieved by creating and integrating a protocol which will assist confidentiality and integrity [19]. The solutions found to create the optimal protocol are as follows:

- The patients must be contacted by the administrators before any use of the medical information is planned. Permission from the patient must also be obtained.
- The identifying information of the patient must be kept encrypted unless specifically requested and permission for this is obtained.
- In the case of emergencies, the next of kin must be contacted for the information. If this is not possible the information must only be released to a select few of the staff working with the patient.

There are many causes of data integrity violations [17]. These can be caused by hardware or software malfunctioning, persons attempting to access the system maliciously and user errors.

Hardware and software errors can be caused by malfunctioning code or sensors in the system. If hardware such as the sensors send incorrect data, this can go undetected as the data may only be off by a few bits. Software bugs can also cause errors in the integrity of the data; this can be detected in most cases, however in some cases this may go undetected for years.

Data management that is trustworthy is currently a challenge to implement, as more data protection methods come out there are more chances that persons with malicious intent will attempt to break those safeguards [4]. With wireless technologies and remote access, the security measures become even more difficult to implement. If an eavesdropper gains remote access to the database, the system may be compromised for an undetermined amount of time as these malicious users do not interfere with the system and are difficult to detect. Data integrity may also be lost through the insertion of a virus into the system, this could be a targeted virus, one which specifically damages the system it enters or an untargeted virus, which could delete all data it comes into contact with, or send all the data in a database to an email address of the person with malicious intent.

User interactions may also compromise data integrity. Users may delete data inadvertently or corrupt the database by having access to a critical section of the system.

There is also integrity assurance techniques which are commonly used to combat the problems determined above.

Mirroring maintains two or more copies of the database in the storage device; the data integrity can be checked by comparing the data to the copies. Mirroring is able to detect data corruption due to hardware errors but cannot provide the correct data as it is not known which copy has the legitimate data. Mirroring does not help to detect users with malicious intent as both copies may be modified by the informed user.

Another commonly used method to perform a data integrity check is a checksum. The integrity of the data can be determined by comparing the stored data and the newly written data every time the data is read.

Cryptographic functions such as hash functions map strings of various lengths to short fixed size results. The functions are designed to be collision resistant; this means that two hash function results will never match.

There are various types of hash functions in use. Message-digest algorithm (MD5) and Secure Hash algorithm (SHA1) use randomness in their properties while keyed hash message authentication code (HMAC) generates cryptographically protected data and is able to detect if a checksum has been tampered with [25].

Authentication of the users with access to the data will need to be implemented in the system. There are many authentication methods used in smart card and NFC systems. NFC systems are currently in use for secure payments; these systems need highly secure methods of authentication [25]. The options currently in use for authentication are password authentication, biometrics, network access authentication, smart card authentication, and IP Security authentication.

For authentication to be successful and secure multiple methods are to be tested and deployed.

2.3.3 Cryptography

Cryptography is used to protect data and information, be it information stored in a single/multiple locations or information being transferred across multiple servers and people [22], [24].

Cryptography is defined as: “a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it.” [17]. Three methods of cryptography were discussed in this section, Data Encryption Standard (DES), Advanced Encryption Standard (AES) and Public Key Cryptography.

2.3.3.1 Data Encryption Standard

Data Encryption Standard (DES) is a symmetric-key block cipher [30] published in 1975 at IBM. DES was designed to encrypt and decrypt blocks of 64-bit data with a key of length 64 bits, as shown in Figure 1. The input key for DES is 64 bits long however the key which is used is only 56 bits in length [26]. The parity bit in the byte is the right-most (least significant) bit; this is set to ensure that every byte contains and odd number of “1s”. The parity bit is however ignored; this reduces the 64-bit key length to 56 bits. The DES algorithm iterates 16 times to compound the plaintext and the key values. This process converts the 64-bit plaintext to the encrypted 64-bit output. Using the same key and the algorithm on the decryption side, the same method can be performed to determine the plaintext from the encrypted text.

It has been found that a key length of 56-bits is too short for the algorithm to be secure [17, 26]. In 1998 the Electronic Frontier Foundation created a computer built to crack the DES algorithm. The computer cracked the algorithm within a matter of days and within the last few years, DES can be cracked within hours using super computers.

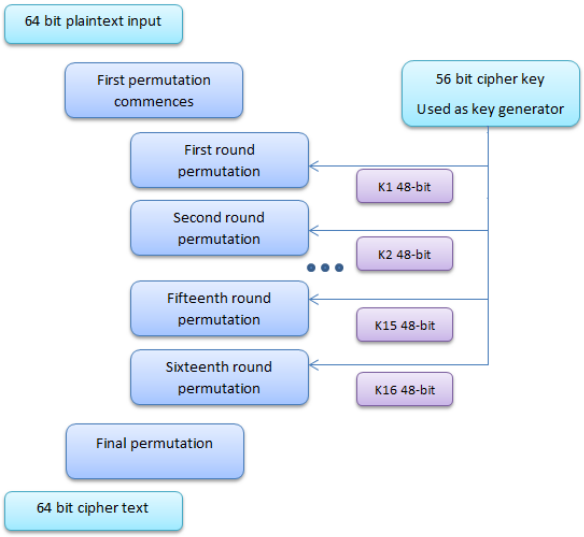

Figure 1: Block Diagram of DES operation [26].

2.3.3.2 Advanced Encryption Standard

The Advanced Encryption Standard (AES) is based on a principle called the substitution permutation network [34]. AES has a block size of 128 bits; the key size can be 128, 192 or 256 bytes as shown in Figure 2.

AES uses a 4x4 matrix of bytes to encrypt the data; the key size determines the amount of transformation rounds of substitution and permutations the plaintext will be cycled through. A few of the steps which the plaintext is cycled through are [26]:

- ShiftRows: Each row in the matrix is cyclically shifted to the left; the number of places the bytes are shifted differs for every row.
- SubBytes: A lookup table is used to replace each byte in the matrix with a value in the lookup table.
- MixColumns: A fixed polynomial is calculated and each column in the matric is multiplied by it.
- AddRoundKey: Each byte in the matric is combined with the sub key byte through the use of the XOR function.

AES was tested and it was determined that at 50 billion key checks per second it would take 5×10^{21} year to brute force attack the 128-bit key [26].

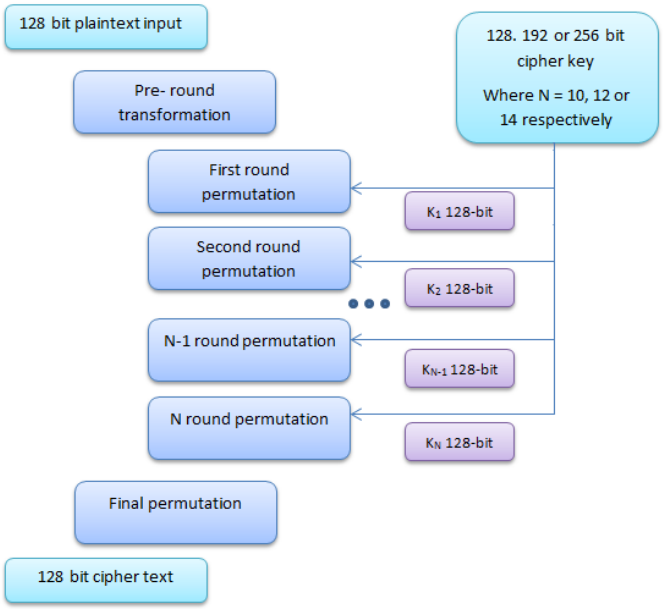

Figure 2: Block Diagram of AES operation [26].

Cipher Block Chaining (CBC) is the cipher mode where each plaintext block is XOR-ed with the previously encrypted block before encryption as seen in Figure 3 and 4. The original block is XOR-ed with the initialisation vector before it is encrypted. The initialisation vector is a randomly generated 16-byte string placed at the beginning of the cipher text.

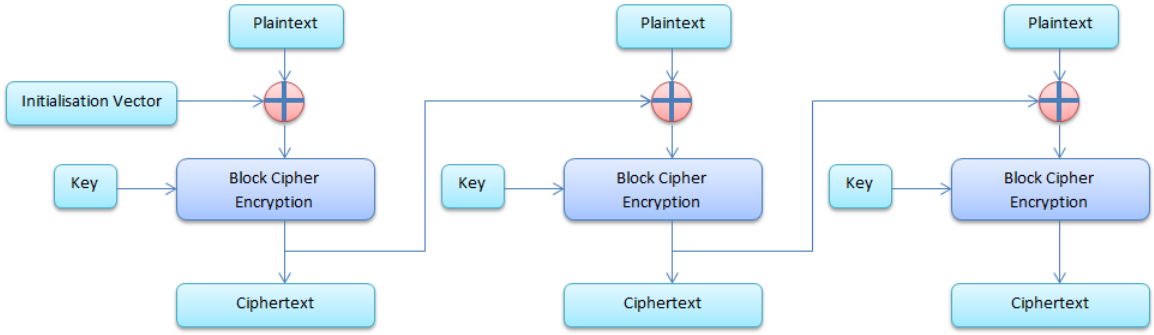

Figure 3: Diagram of CBC encryption [26].

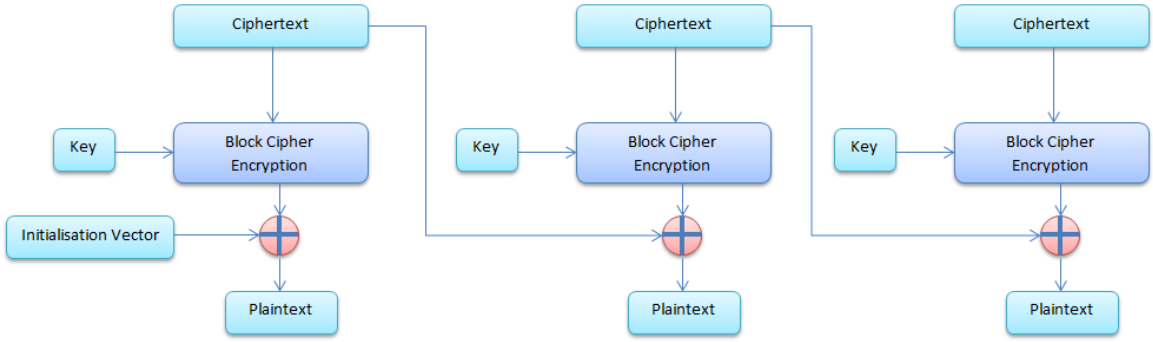

Figure 4 Diagram of CBC Decryption [26]

2.3.3.3 Public Key Encryption

Public key encryption uses an asymmetric-key pair, a public key and private key for encryption and decryption [36]. The public key is distributed freely and is publically known, the private key is kept a secret to the person encrypting/decrypting the data as shown in Figure 5. A public key can only be decrypted with the corresponding private key and the private key can only be decrypted using the public key. In addition to the public and private key a digital signature can be used. A digital signature authenticates that a message comes from the source which it claims it does [24].

The public and private keys are generated as follows [24], [26]:

1. Any two prime numbers, p & q are selected
2. The value for n = p*q is calculated
3. Euler's totient function Φ is then calculated. Φ = (p-1) *(q-1)
4. Calculate the values for d and e where d*e = 1 mod Φ
5. The public key is then e and n and the private keys are d and Φ

Certification Authorities (CA) provide proof of a person or organisations identity. The CA signs and delivers a certificate which assures the sender that the signed public key belongs to the person the public key holder claims to be. If the third party CA is trusted is can be assumed that the public key holder is also trusted as a certificate, once issued cannot be altered by other sources.

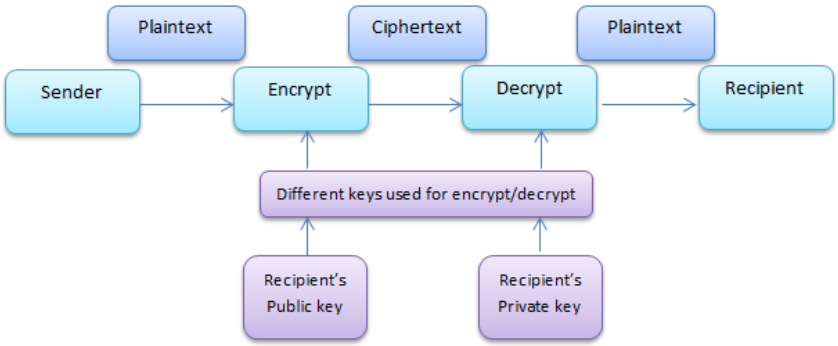

Figure 5: Diagram of Public Key encryption operation [26].

2.4 MANAGING MULTIPLE MEDICAL DEVICES

The number of medical devices in one hospital or clinic can be in the hundreds [14] and an efficient method of managing and identifying these devices is needed for accurate data logging.

In a system with multiple sensors, the exact location of the sensors directly affects the lifetime of the network. Location information and placement of sensor nodes needs to be collected and analysed by looking at the following points pertaining to a sensor network in a farmer's field [15]:

- Placement of the sensor nodes - The position where the sensors are placed effects the data retrieval, the sensors need to be able to connect to the wireless network at all times.
- Location information of the nodes - An estimation of where the sensor node is located is needed for accurate data recording.
- Collection of the data - A device will be needed to collect the information on a regular basis from the nodes.

In addition to the positioning of the sensor nodes in order to manage the sensor network, there are other factors which need to be determined depending on the environment [8]. These factors determine the reliability of the wireless network.

- Longevity: Any operations in the wireless sensor network need to be as energy efficient as possible, sensing data should only be passed to the application when it is needed.
- Latency: The time between the data being delivered to the application and the data being recorded by the sensor is the latency.
- Accuracy: The data retrieved from the sensor needs to be reliable and exact. Accuracy can degrade with a higher latency or certain environmental factors and these factors need to be accounted for.
- Fault tolerance: The network needs to continue operating even in the event a node failing, communication problems or physical destruction of a node.

A system needs to be implemented to include all the factors listed, for the health sensor system to work reliably and efficiently.

2.5 MEDICAL DATA LEGISLATION

The security of medical data is a major concern as outlined in Chapter 2.3, however in addition to these security measures implemented by various other applications, the legislation outlined by the South African Government and the Personal Protection of Information Act (POPI) are taken into account. This section outlines this legislation and optimal methods to adhere to them.

A health record as defined by the Health Professions Council of South Africa [26] is any relevant record made by a health care practitioner at the time of or subsequent to a consultation and/or examination or the application of health management.

A health record contains the information about the health of an identifiable individual recorded by a health care professional, either personally or at his or her direction.

The health record in the case of this dissertation is the patient medical data entered into a smartphone application by the professional and uploaded to the database of all patient medical data within South Africa.

Record maintenance is kept to very high standards within the medical profession [27]. The list below details the record keeping rules which apply to the medical database and application.

1. No information or entry may be altered or removed from a medical health record.
2. An error must be corrected by placing a line through it and with ink and correcting it.
3. Additional entries must be dated and signed.
4. Any errors or amendment must have a valid reason for the change attached to the error.
5. Health records should be stored for a period not less than 6 years from the date they become inactive or dormant.

Access to personal medical records has multiple regulations surrounding it. These regulations need to be applied in the case of personal access to their medical health records.

1. A health care practitioner may provide direct access to a person 12 years and older to his or her own records on request (Children's Act No 38 of 2005).
2. A person under the age of 16 years may have a parent or guardian request access to the medical records; however this is only on written consent from the patient. (Access to Information Act No 2 of 2000).
3. No information may be made available to a third party without the written authorisation of the patient or by court order. (National Health Act No 61 or 2005).

The South African National Health Act outlines the protection of medical data records by using the following guidelines [28]:

1. The person in charge of the health establishment which has possession of any health records must have control measures in place with prevent the unauthorised access to the medical records and the storage facility in which they are kept.
2. Records must also be protected from damage, in particular digital records should be backed up regularly and the backup should be kept in a separate off-site location.

For a medical database to store patient health records, the guidelines outlined in this section are adhered to strictly.

The Protection of Personal Information (POPI) Act was created to ensure that South African institutions are responsible when collecting, storing, processing or sharing other entities personal information [27]. This Act prevents the abuse of personal information in any way. As the personal information outlined in the Act includes the information collected by the medical health device, the following precautions are taken into account while designing the application and database:

1. Personal information can only be collected from the owner of the information
2. The owner of the information must be informed when the information is collected
3. The reason for the collection of the information must be valid
4. The reason for the collection of the information must be transparent with regard to the purpose and intended use of the information
5. The information must only be shared with authorised parties
6. The information should be maintained and records of use should be complete
7. Owner of the information collected should have access to it when requested within a reasonable time period
8. The owner of the information should be notified if any security measure to protect the data has failed
9. All policies and procedures relating to the protection of the data should be reviewed and updated regularly.

2.6 CURRENT MEDICAL SYSTEMS

Within South Africa and rural Africa there are systems which have similar outcomes to the work proposed in this dissertation. These systems aid in the evaluation of the designed and implemented solution. Two systems are briefly discussed in this section. In Chapter 5 the designed solution from this dissertation will be compared to the solutions presented here.

2.6.1 Health Information Exchange

Health Information Exchange (HIE) is a solution developed by the CSIR for Rwanda and currently being adapted for use within South Africa [31], [32], [33]. The HIE is a generic

architecture which includes a Public Health Record (PHR) for each citizen of the country. The HIE takes into account design considerations which could affect the system within South Africa, the considerations are listed below.

1. Interoperability – This includes the ability of information and communication technology systems to communicate to the business and processes which they support. The multiple medical systems within South Africa need to become interoperable to ensure that the creation of an effective national electronic, patient based information system. This can be completed through the integration of multiple systems currently available and where possible.
2. Comprehensiveness – The medical health record needs to be comprehensive, up to date and the information should be from a trusted party.
3. Legal Value – The patient should be in control of the record, the ability to grant or deny access to external parties should be controlled by only the patient or authorised caretaker.
4. Availability – The medical records should be easily available when a healthcare provider requires it.

2.6.2 mHealth

The South African National Department of Health has created the strategy called Mobile Health (mHealth) [34], [35]. mHealth is defined a subset of eHealth which allows for the delivery of health related services via mobile communications technology. This includes services such as emergency response services, disease surveillance and control, remote patient monitoring and control, health related m-learning and synchronous and asynchronous mobile tele-medical diagnostic. One of the implementations of mHealth is the MomConnect system, created in August 2014. MomConnect uses a smartphone application to allow pregnant women to register themselves in antenatal care facilities. The system sends out eHealth messages to improve the health of themselves and their babies. This also includes monitoring of the pregnant women during and after the pregnancy and the exchange of medical information between facilities.

The relevant challenges encountered in the mHealth system are listed below:

1. Lack of interoperability

2. Absence of a single framework
3. Lack of use of open-source options
4. Absence of practical approaches to privacy and security.

2.7 CONCLUSION

The literature study covers the documentation needed to accurately design and build the medical sensor device with regard to the transmission of data, security of data and relevant legislation surrounding the collection and storage of the data.

The literature study assists in reviewing the wireless technology available for the use in the medical health application for various implementations. Bluetooth, ZigBee, Wi-Fi and NFC/RFID were investigated for use in the application.

Available security methods are reviewed and the best option for implementation in the dissertation is detailed. It is seen that both Public Key encryption and AES encryption are viable solutions for implementation and that DES encryption will not be used as the implementation is insecure with recent developments in technology.

The presence of similar systems in a South African context is analysed for any considerations which can be applied to this dissertation. The system presented in the dissertation is compared to the similar systems in Chapter 5.

CHAPTER 3 METHODS

3.1 CHAPTER OVERVIEW

This chapter covers the architecture and implementation of the patient-based monitoring system. The chapter covers four main topics:

1. The proposed system configuration,
2. the transfer of medical information between the device and the smart phone application,
3. the NFC token based authentication of the data transmitted to the database, and
4. the optimal cryptographic design for the data transfer and throughput.

Chapter 5 contains a discussion on the adherence of the system to current medical health record legislation.

3.2 PROPOSED SYSTEM CONFIGURATION

The proposed system configuration is detailed in Figure 6 below; the full system description will be detailed in the following sections.

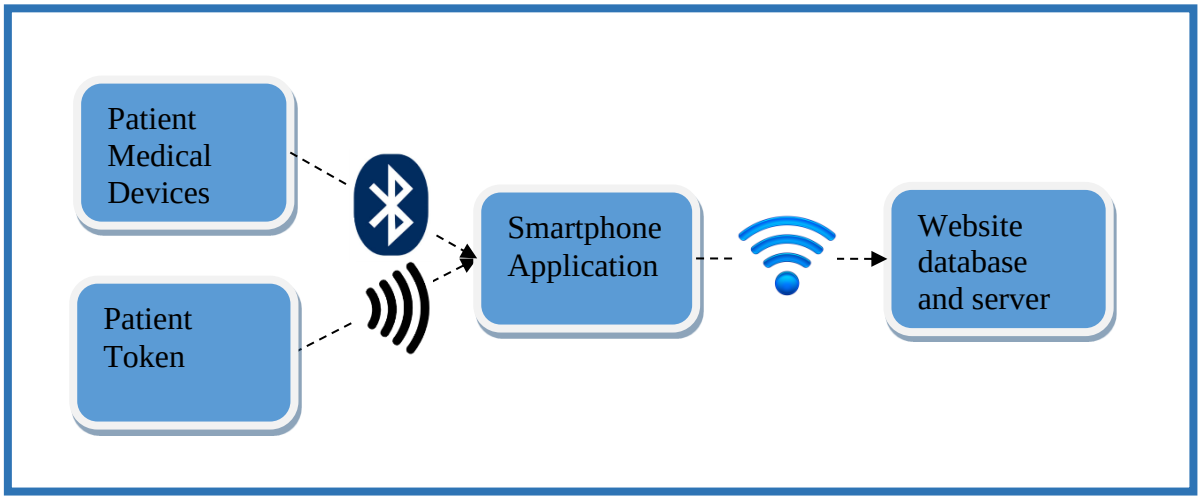

Figure 6: Block diagram of the proposed system design

The proposed system configuration allows for patients and medical practitioners to measure vital signs of patients and then upload the vitals using Bluetooth to a smartphone application which securely sends the data to a global database. The global database

contains all personal information on the patient such as ID number, name, allergies, medical aid as well as previous medical tests, screenings and vital signs completed through the current medical practitioner as well as past medical practitioners. Patients have limited access to the application allowing only vital sign updating and personal information access and updating. Access is granted to the application through a personal medical card. The medical professionals are allowed complete access to a patient's database only once the patient grants access through his own medical card. The system configuration takes into consideration all the security aspects surrounding the application In addition the dissertation documents and details all the legislation and laws surrounding the transfer and use of digital patient medical and personal data.

3.3 TOKEN BASED AUTHENTICATION AND SMARTPHONE APPLICATION

A security token can fall under one of three categories: a disconnected token, a connected token and contactless token. The NFC card is an example of a connected token. A disconnected token will have no physical or logical connection to the Smartphone application and an example of a connected token is a USB token [27]. The NFC card is used as a security token for the access to the medical data; the card establishes the identity of the patient and is used by the medical practitioner to connect to the patient's medical data stored on the medical database. The NFC card is the property of the patient and only contained an encrypted key. For speed comparison purposes a disconnected token is created using an Arduino Uno program and graphical display, the token displays a password for each application login which the user needs to manually enter.

3.3.1 Smartphone Application Structure

The smartphone application is designed using data confidentiality as the main consideration. When the application is opened by the user, the main screen appears, this is the log in screen which can be used for both the medical practitioner and a general user who needs to update/view their medical information.

The smartphone application has two components. The first component, the Practitioner component, is built for the medical practitioners. The second component of the smartphone

application is created for all citizens with the medical NFC card and labelled the Patient component.

3.3.1.1 The Practitioner Component

The Practitioner method allows only certified medical practitioners to access the application. This is done through the NFC card authentication. A medical practitioner who is active in the medical field will be granted an NFC card which authorises this. Once the application is opened the practitioner logs on to his application using his own NFC card, he can then either log onto a patient's database using the NFC card given to him by the patient or access his personal portal. The medical practitioner is allowed to be logged in for one hour before being automatically logged off and requested to enter his password once again; this protects the data if the phone is lost or stolen while logged in. The session can be ended by the patient once the consultation with the medical practitioner is over. In Figure 7 the method for the medical practitioner to access his/her personal portal or the patient medical database is described.

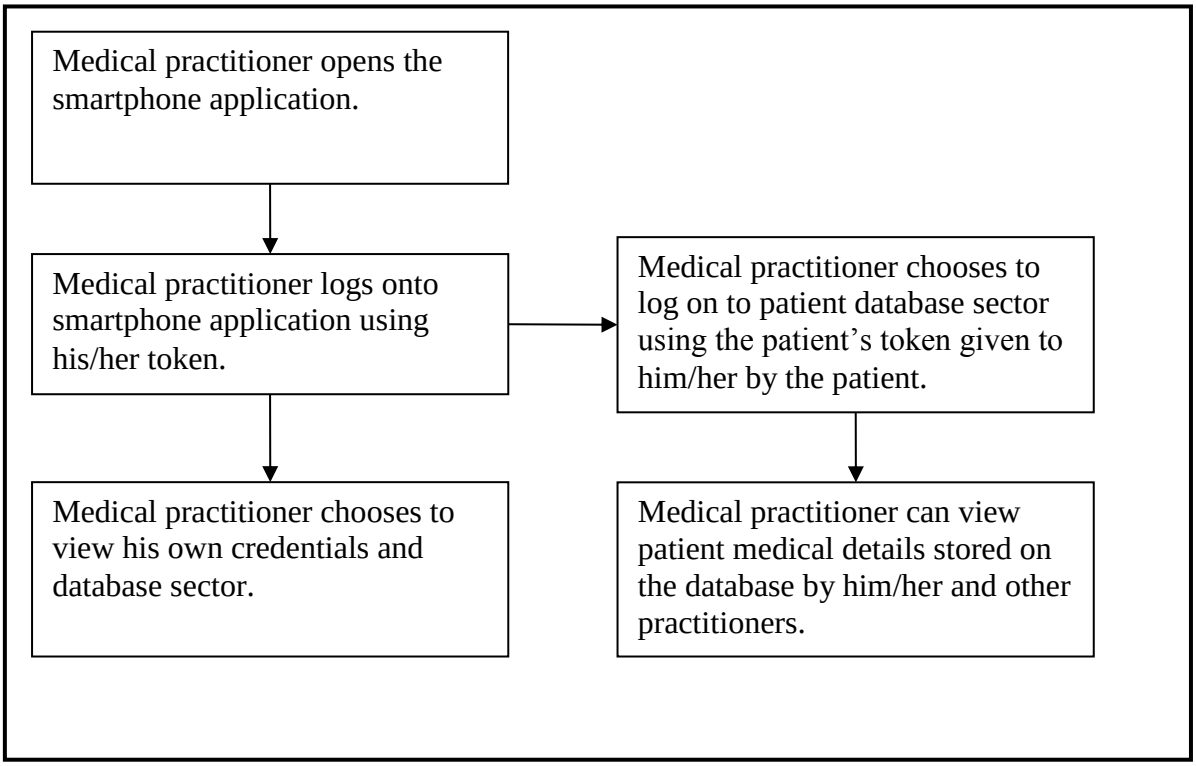

Figure 7: Figure displaying the medical practitioner's component of the smartphone application.

3.3.1.2 The Patient Component

The Patient token is granted to a patient when a medical database sector is created for the person, usually at birth. Access to the database is limited to the medical practitioners of the patient and can only be granted by the patient through the token authentication. When the patient logs on to the application using their token, the personal portal is opened. This allows the patient to update any personal details such as emergency contacts and addresses. The patient is also allowed to upload blood pressure, blood glucose and heart rate readings as described in Chapter 3.4. However, the patient component does not allow the adjustment of medical records; this can only be done by a registered medical practitioner due to legislation surrounding the protection of health records. The patient portal only allows a login session of half an hour for data safety and security. In Figure 8 the method for the patient portal access is described.

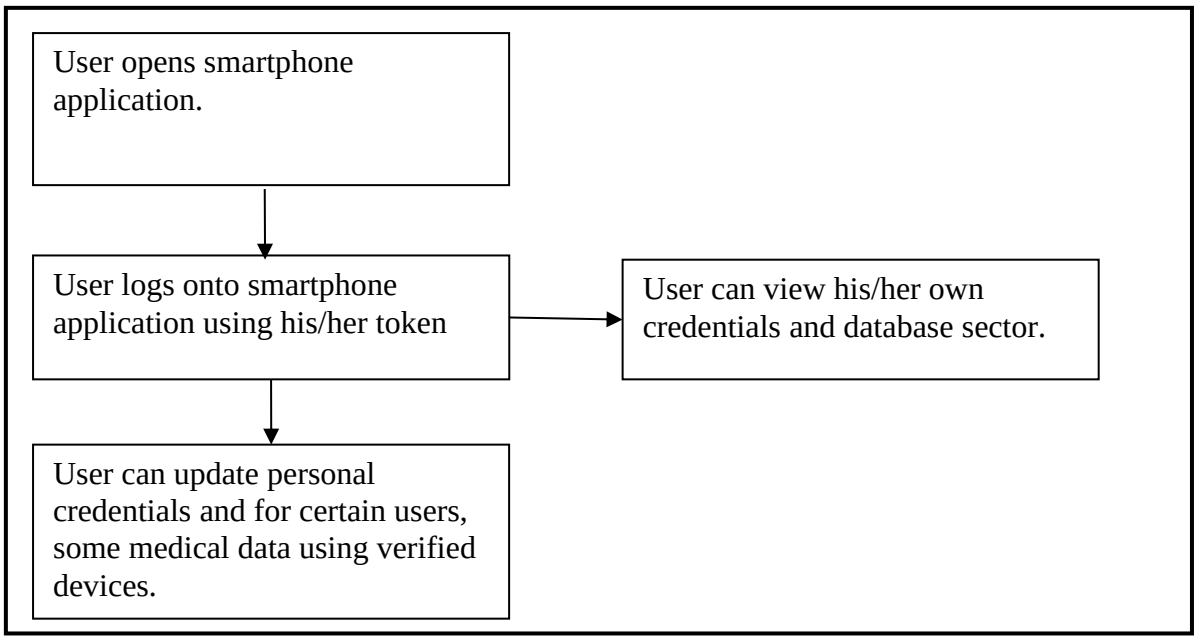

Figure 8: Figure displaying the average user's component of the smartphone application.

3.3.2 NFC data transfer protocol and authentication

Multifactor authentication (MFA) is a security system that requires multiple methods of authentication [26]. This application's main focus is on the protection of medical and personal data protection, so MFA is applied in various ways within the application to allow for the security of the data. The NFC card is utilised in two ways within the application. The first method is when the NFC card carried by a general user is used as a hardware token to provide the first layer of security to the patient's medical data. A general user of the application needs the NFC card to log in to the Patient component of the application. The NFC card contains the original hashed password which was initialised on NFC card when the user was issued the medical NFC card. When the NFC card is pressed against the phone the user was prompted to enter a username and password, this username and password is then matched to the hashed password stored on the NFC tag. If the passwords match, then the application is issued a token containing the users hashed ID number. This token is then sent to the online database for verification. Once this is verified to be the same as the application users, the portal is then available.

The medical practitioner follows the same steps to log in to the Practitioner component of the application; however, when access is needed to the medical data of the patient the NFC card of the patient is required once again. Figure 9 shows the NFC card used in this situation.

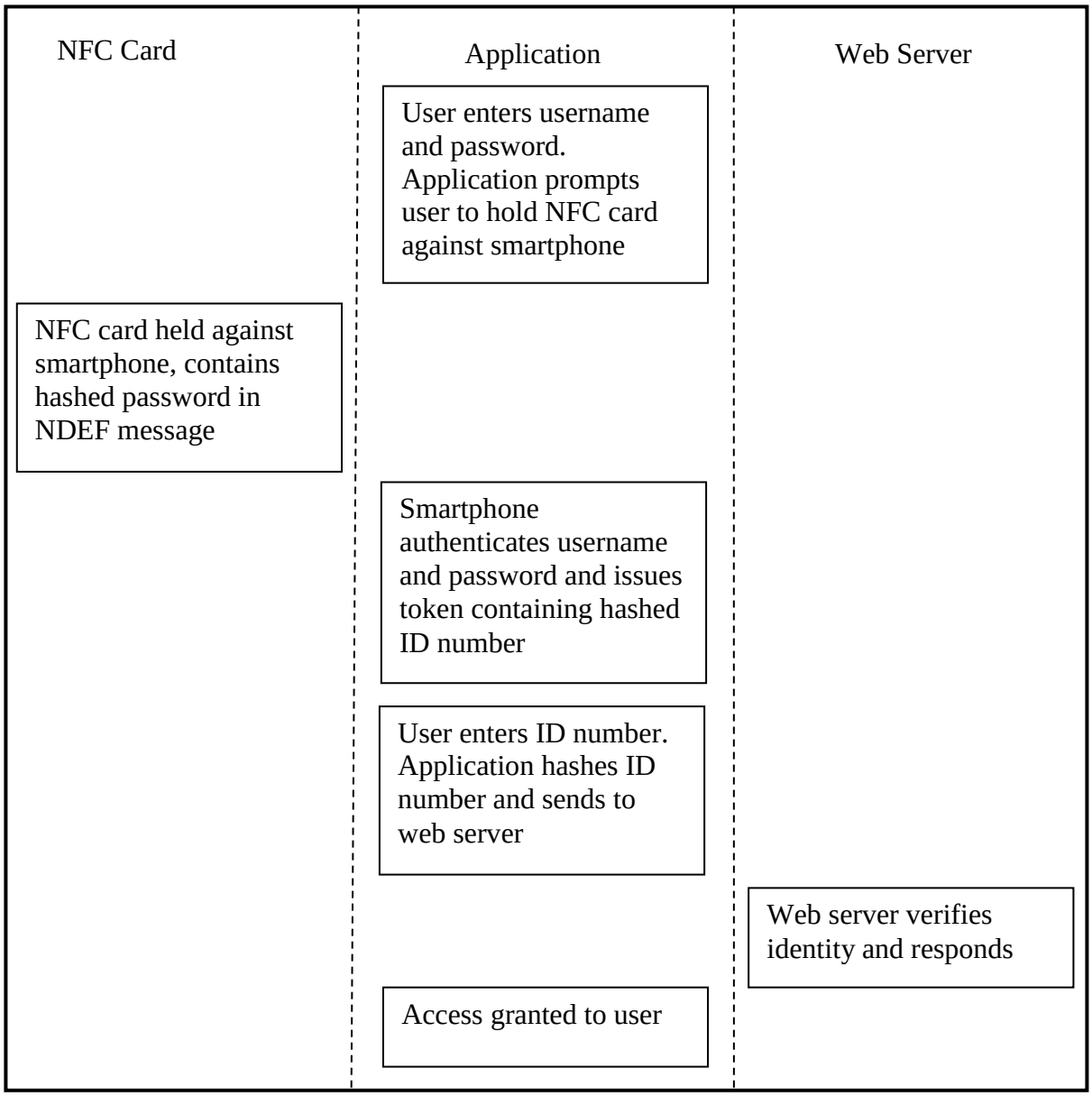

Figure 9: Figure displaying the background operations when the application runs

The second use of the NFC card is for the access to the medical data. When the medical practitioner requires the medical data of a patient; the patient hands him/her the NFC card. The practitioner then holds the NFC card to the smartphone. The application verifies that the practitioner has permission to access this patient's information by digitally signing the request for information with the patient's private key and generating the signature using

the previously established password. The digital signature is encrypted and sent to the medical record web server. The web server verifies the information using the public key.

The patient is notified through his application of the medical practitioner's access of his/her medical information. When new information needs to be uploaded to the database the medical practitioner once again needs to use the NFC card to digitally sign the data before it is sent. This adds a layer of security to the application which protects both the medical practitioner's information as well as the patients. In Figure 10 the NFC card usage in this manner is shown, Figure 9 occurs before the events in Figure 10. In addition, the medical practitioner may access the patient data when the patient is not available through use of the application. The patient may grant the medical practitioner access to the medical data for a set amount of time; this can be set to years, months or days.

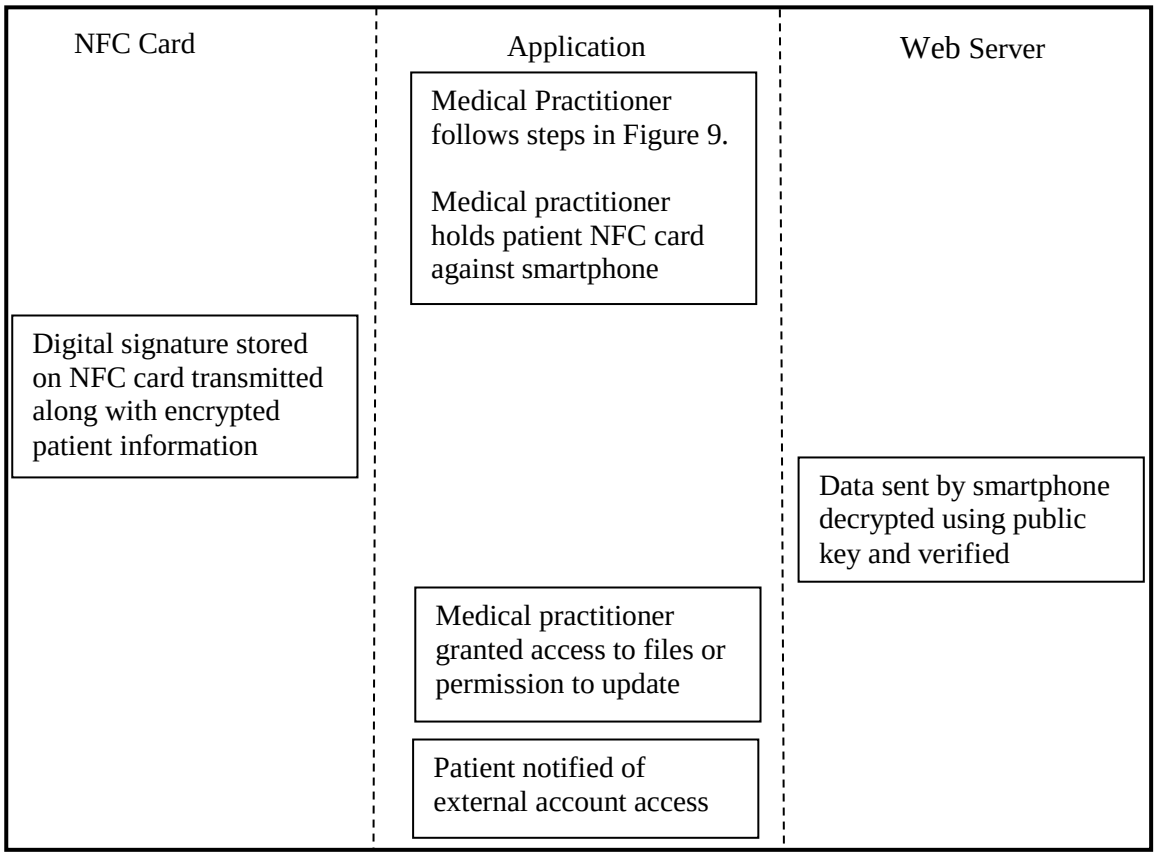

Figure 10: Figure displaying background operation when medical practitioner requests patient information or to update

The NFC card is the first stage of the MFA while personal credentials are used in the second phase of the authentication. In future implementations a one-time password authentication method may be added.

A protocol was designed for the data transfer between the NFC card and the smartphone application. The protocol uses NFC Data Exchange Format (NDEF) messages as the basis for the design. An Android smartphone reads and transmits NDEF data to and from an NFC tag. For this application a separation was needed for the data stored on the NFC card between the password authentication and the digital signature.

Once the application detects the NFC tag being held adjacent to the smartphone, one of two methods are called: the password method or the digital signature method. The method is chosen by the operation which is currently being performed by the application's user.

The application reads all the data on the NFC card, to separate the digital signature from the encrypted password, the application uses the pre-set identifiers.

When the digital signature method is run the application encrypts the message and signs the encryption with the NFC tag used as the digital signature token.

3.3.3 Disconnected token authentication

As described in the NFC token section, MFA is used with the disconnected token to provide security for the patient data. The disconnected token works as seen in Figure 9, similar to the NFC card.

The disconnected token was chosen for the advantage that the user will not need a computer to connect to the application as with connected tokens.

The Arduino Uno contains a program which randomly generates a one-time password for the user's login for that session. The 12-digit password is generated using the random() function. The password contains characters such as A-Z, a-z, 0-9 as well as special characters. The password is then hashed and sent via Wi-Fi to the database, which stores the hash value.

The user enters his username, password and the 12-digit password to the application which is displayed on the LCD screen attached to the Arduino. The application verifies the username, password and the hash value of the 12-digit password and allows the user

access. The system then works in the same manner as described in Figure 9 with the NFC card being replaced by the disconnected token.

When a doctor needs to access a patient's medical database, the process in Figure 10 is followed, however with the disconnected token replacing the NFC card.

For future applications the disconnected token, if deemed to be more reliable or secure than the NFC card, may be created using a smaller interface, such as a key fob.

3.4 CRYPTOGRAPHIC DESIGN

When sending patient information over the internet to the webserver database, there are multiple security considerations. As outlined in Chapter 2.5 in the literature review, the health record data protection regulations are applied. To implement the legislation, various encryption methods are tested and evaluated.

To ensure that the data received by the webserver is from the correct source, a digital signature is sent. This method uses public key cryptography. The second method tested is the Advanced Encryption Standard (AES).

3.4.1 Public Key Encryption

As outlined in detail in Chapter 2.3.3, public key encryption uses an asymmetric pair of keys, a public and private key. The public key is owned by the medical database webserver and it is utilised when a patient sends or retrieves data from the database along with the digital signature. The steps for the encryption are as follows and can be seen in Chapter 2.2.3, Figure 3. For simplicity the digital signature section which has been described in Chapter 3.2.2 has been excluded.

1. A user of the application requests access to the database to send confidential patient information.
2. The application calls an instance of the KeyPairGenerator function and initialises it with a 2048-bit key size.
3. The genKeyPair() function is then called which generates privKey and pubKey.
4. The public key is sent to the webserver which knows to wait for the incoming data.

5. The patient data is then encrypted using the private key and transmitted to the webserver. The webserver knows to check for the digital signature before storing the data on the patient database.

3.4.2 Advanced Encryption Standard (AES)

To store and transmit the user's data securely, the data cannot be sent in plaintext. Using the studies presented in the literature, it was decided that AES encryption is the most reliable method to use. This allows the user to have the data sent over the internet to the webserver securely.

The algorithm chosen for the AES encryption uses cipher block chaining (CBC) and PKCS#5 padding. The key length used is 256 bits as seen in Chapter 2.2.3 Figure 2.

1. To encrypt the data for the first time the user is requested to sign in. The application then runs the request.
2. The application creates a random salt, and initialised the secret key and the initialisation vector.
3. Using these strings, the cipher is initialised and the function output the encrypted data using 1000 iterations as explained in Chapter 2.3.3.2, which is then sent to the webserver along with the salt, initialisation vector and the number of iterations.
4. The ID number and digital signature is also sent to the webserver.
5. When the webserver needs to decrypt the data the webserver verified the ID number and digital signature and sends the password requested to the decrypt function.
6. The application has already grabbed the initialisation vector, salt and the iterations from the string.
7. The method for decryption is then run; however it is run in reverse to the encryption method using the known salt, IV and iterations which output the plaintext data to the correct location on the database.

3.5 DEVICE COMMUNICATION

The transfer of vital signs using consumer devices is completed through the use of Bluetooth. Currently there are multiple medical devices which can transmit readings via

Bluetooth to smartphone applications. For the purposes of this research the following three devices were used:

- Fora IR20 ear thermometer [28]
- Fora P20b Blood Pressure monitor [29]
- Fora G31 Blood Glucose monitor [30]

These devices measure the patient's temperature, blood pressure, pulse and the blood glucose levels. The devices are also able to transmit the readings via Bluetooth to a smartphone. This communication is shown in Figure 11.
In addition, for comparison of the Bluetooth transfer speeds an Arduino Uno with a Bluetooth Bee Bluetooth module was built and coded to transfer dummy medical data. This device is used to test the viability of the off-the-shelf medical devices when compared to a device which records the medical data from normal medical devices and sends the data using a Bluetooth module.

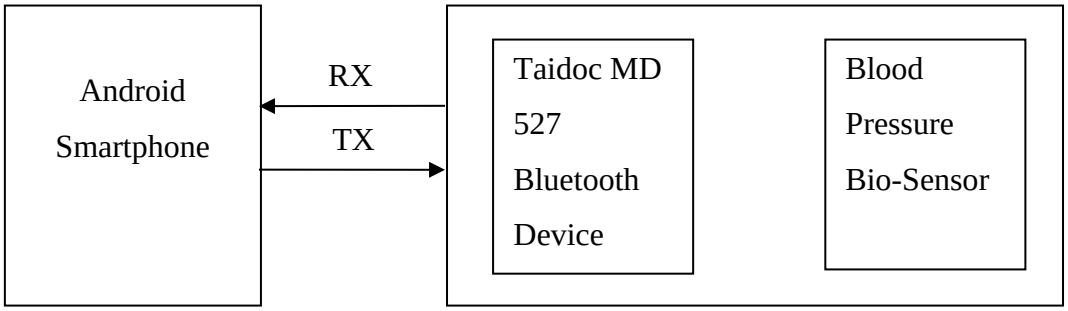

Figure 11: Figure showing the communication between the Android smartphone and the Bluetooth medical devices

To communicate with the Bluetooth devices there are four main components: Android Bluetooth Permissions, device queries, device discovery and device connection.
Once the application was written and the permissions set up, the built-in Android Bluetooth Adapter function is used to connect with each device. The application utilises the following steps to establish a connection:

1. Call default Bluetooth Adapter which determines if the device has Bluetooth functionality
2. Determine if Bluetooth module is switched on and if not ask user permission to switch module on
3. Scan area for available Bluetooth devices and display using ListView
4. Ask user to select correct device
5. Establish connection

To communicate with the Bluetooth devices, the datasheet for each device was requested from Fora.
Utilising the datasheet it was determined that the medical devices are set up to work as a client while the smartphone acts as the master, retrieving stored data from the device. For each medical device there is a different method to access the data stored on the device as described further on in this chapter. However; for the communication between the application and the medical devices the method is as follows.

To share the data on each device between the Bluetooth socket, an InputStream() and OutputStream() method is set up to handle the transmission of data. A dedicated thread is used for all the reading and writing done with the application; this is done by writing all the Bluetooth communication in a separate Android activity and then linking the activity to the main activity. The reading and writing to the stream is done by calling the functions read(byte[]) and write(byte[]). Read(byte[]) blocks the input stream until there is something to read while write(byte[]) does not block until there is not enough space in the buffers, which results from too much information being sent by the medical device.

The setup for the application is as follows: The constructor sets the streams and executes them once the connection has been made, the application then sends the command needed by the medical device to the stream.
The thread then waits for the data to come through the input stream and sends the data back to the main activity by the use of a Handler initialized in the parent class. Once this data is sent the thread waits for more input or output from the device and application. Once

the application has received the data needed the application then closes the stream so that it is free to be used for the next device.

3.5.1 Ear thermometer communication

The thermometer used in the dissertation is the Fora IR20b Ear Thermometer [28] seen in Figure 12. The ear thermometer acts as a slave device and once connected waits for commands received from the master device, the Android smartphone.

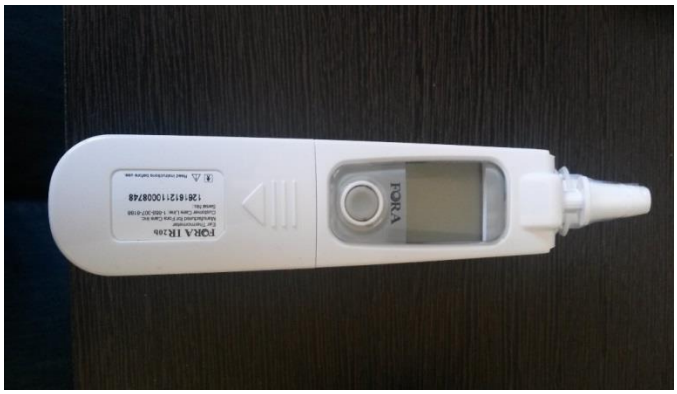

Figure 12: Fora IR20b Ear Thermometer

The thermometer uses an 8-byte UART command to fetch data from the thermometer. The Android smartphone sends the command for the data via the TX port, the thermometer responds to the smartphone with a hexadecimal value representing the temperature through the RX port. The temperature readings are returned in units of 0.1 ºC and the application converts that value before sending the data to the patient database.
The frame structure of the commands to the thermometer is shown in Table 2. This is the format used to send the commands to the thermometer. The Data_0 field is the hexadecimal value 26 according to the datasheet of the thermometer.

Table 2 Table showing the 8-byte frame structure for thermometer communication

Byte Direction	1 Start	2 CMD	3 Data_0	4 Data_1	5 Data_2	6 Data_3	7 Stop	8 ChkSum
MµC → DµC	51	CMD	Data_0	Data_1	Data_2	Data_3	A3	[1..7]

MµC ← DµC	51	ACK	Data_0	Data_1	Data_2	Data_3	A5	[1..7]

3.5.2 Blood Pressure monitor communication

The Blood pressure monitor used for this dissertation is the Fora P20b blood pressure monitoring system [30] as seen in Figure 13. The blood pressure monitor acts as a Bluetooth slave device and once connected awaits commands from the master device, the smartphone application.

Figure 13: Fora P20b Blood Pressure monitor

The blood pressure module operates in a similar manner to the ear thermometer. An 8-byte UART command fetches data from the monitor. The Android application sends an 8-byte command through the TX port and the blood pressure monitor sends back the result using the RX port.

The command to the blood pressure monitor is first sent to retrieve the diastolic blood pressure value; it is sent back to the smartphone application as a hexadecimal value and converted to decimal value.

The second value retrieved from the blood pressure monitor is the systolic blood pressure value. This value is also sent back as a hexadecimal value and converted to the displayed decimal value.

3.5.3 Blood Glucose monitor communication

The Blood glucose monitor used for this dissertation is the Fora G31 blood glucose monitoring system [31] as seen in Figure 14. The blood glucose monitor acts as a Bluetooth slave device and once connected awaits commands from the master device.

Figure 14: Fora G31 Blood Glucose Monitor

The blood glucose module operates in a similar manner to the ear thermometer and blood pressure monitor. An 8-byte UART command is used to fetch data from the monitor. The Android application sends an 8-byte command through the TX port and the blood glucose monitor sends back the result using the RX port. The result is received as a hexadecimal number and was converted to a decimal number before transmission.

3.5.4 Arduino Uno and Bluetooth Bee communication

The Arduino Uno does not have built-in Bluetooth functionality however there are devices which can be attached to the Uno for Bluetooth support. The XBee shield allows modules such as the XBee and the SeedStudio Bluetooth Bee to be attached to the Uno as seen in Figure 15. The XBee shield was connected to the Arduino Uno along with the Bluetooth Bee.

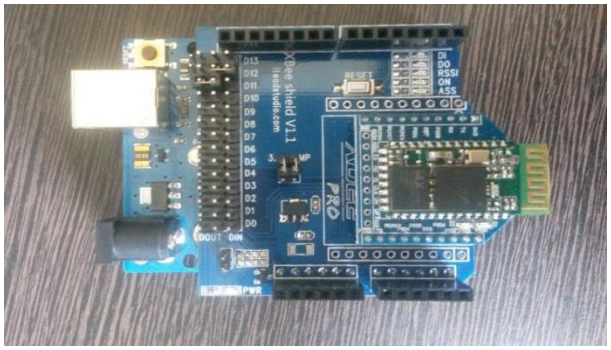

Figure 15: Arduino Uno with XBee shield and Bluetooth Bee attached

The Bluetooth Bee is a SPP (Serial Port Profile) module which is compatible with XBee shields to connect to the Arduino Uno. The Bluetooth Bee uses Bluetooth V2.0 with Enhanced Data Rate of 3 Mbps, a 2.4 GHz radio transceiver and baseband modulation [17]. A description of the hardware setup for the Bluetooth Bee is seen in Chapter 2.2.4. To communicate with the Bluetooth Bee the Software Serial library (SSl) was used. The SSl allows for serial communications using the RX and TX ports of the Bluetooth Bee. The Bluetooth Bee communicates at a default baud rate of 38400 and a series of AT commands (a series of short test strings used mainly with routers) were used for setup and communication.
The medical devices described in the sections above operate as slave devices, only sending data via Bluetooth. To simulate the devices, the Bluetooth Bee was set up in slave mode.

The first step in setting up the Arduino application was to define the RX and TX ports corresponding to the correct location on the Arduino Uno Xbee Shield. The ports chosen were ports D12 and D11, D12 was connected to DIN which sets it as the TX port while D11 was connected to DOUT setting it as the RX port. The next step was to setup the Bluetooth connection; this was done by sending the same AT commands as described previously. Since the baud rate at which the Bluetooth Bee operates is 38400, the serial port was set to 38400 by the Software Serial begin() function.
Once the setup has completed and all the commands have been executed the program goes into the loop() function. This is the function which the Arduino Uno loops through operations continuously after the setup() mode. The loop() function first determines if the

serial port is available, that there is currently no communication ongoing between the Bluetooth Bee and the Arduino. If the port is unavailable, it keeps looping through until the port is available. Once the port is available the Arduino waits for input from the smart phone, the input chosen for the smartphone to write was 0x26 for continuity with the other medical devices.
The Arduino is set up to send dummy medical data to the smartphone once the smartphone connects.

3.5.5 NFC card and Android communication

3.5.5.1 NFC read() function

The read() function is the function which reads the data from the NFC tag using the on-board hardware of the smartphone.
When an NFC tag is detected the Android device executes the following steps:

1. Parses the NFC tag and finds the MIME type or the URI which identifies the data payload in the tag. The MIME type is the identifier for the file format while the URI is the uniform resource identifier, which is the string of characters used to identify the NFC tag, this is expanded on further below.
2. Encapsulates the MIME type or URI and the payload into an intent.
3. In a well formed NDEF message, the NDEF record should contain the following fields, where each field has the type used in this dissertation with a short description:

    - 3-bit TNF (Type name format) -This section indicates the variable length type field. For this dissertation TNF_WELL_KNOWN is used. TNF_WELL_KNOWN is when the MIME type or URI is dependent on the Record Type Definition (RTD), which is set as RTD_TEXT, a simple text field.
    - Variable length type - Describes the type of record, this is the field where RTD_TEXT is defined.

- Variable length ID - This is the unique identifier for the record, each tag needs a unique identifier and for this application the identity number of the patient is used. This is due to the fact that when linking the data to the database the information will be stored in the correct format already.
- Variable length payload - This is the actual content of the record for this dissertation, the patient's details.

Once the tag dispatch system has used the TNF and type field to map the MIME type or URI to the NDEF message the information is encapsulated inside the ACTION_NDEF_DISCOVERED intent along with the payload. If the tag dispatch system cannot determine the type of data in the record i.e. when there is an error in the record, the payload and tag data is encapsulated in the ACTION_TECH_DISCOVERED intent and is not a valid tag for the use of this dissertation. In this application, when the intent is discovered the ACTION_NDEF_DISCOVERED is called. This starts the main Activity of this dissertation. If an ACTION_TECH_DISCOVERED intent is called the application brings up a message which states that the tag is not valid and to try again. This is also brought up when the tag is removed before the intent can be discovered. Another function which the application opens is ACTION_TAG_DISCOVERED. This is for a new tag with no data stored on the tag, and allows the user to write to the tag but not to read from the tag.

This is the basic outline of how the smartphone interacts with the NFC tag. For any of this interaction to begin the application needed to have permission to use and exploit the NFC reader/writer on the smartphone. This was done by requesting access to NFC in the Android manifest. Done by the following line of code:
<uses-permission> android:name="android.permission.NFC"

For the read() function to operate efficiently, a few checks were done before reading the NFC tag.

1. The application first checks if the smartphone it has been installed on has NFC capabilities. If not, the user of the smartphone is notified of this through a popup

message. The application continues onto the next step if NFC capabilities are available.

2. If the smartphone does not have the NFC functionality turned on the application cannot use the hardware. In this case the smartphone checks if NFC is enabled, if not, a message appears informing the user that NFC is disabled and requests if the user would like the application to turn on NFC. The application continues on to the next step once NFC is activated.

Once the read() function is called by the user the following steps are followed:

1. The function checks if an NFC tag has been placed against the back of the phone, if there is no NFC tag, the user was informed to hold the tag to the back of the smartphone.
2. The function then verifies the encoding used on the tag is MIME or URI type encoding or the application will not be able to run the function. If this is the case the user was required to use another tag and is informed of this.
3. Once this has been verified the NFC tag data is read and the login in or transfer data function is called.
4. To read the NFC tag data the function connected to the NFC tag by getting an instance of NDEF from the tag and storing the result to a string. Using this result, the smartphone can now connect to the tag using the NDEF.connect() function. Once this has been set up the function checks if there is any data stored on the tag or if the tag has no data stored to it. If there is no data stored to the tag a message is displayed. If there is data stored to the tag the data payload is read from the tag and stored to a string. The string then displays on the smartphone screen to the user.

3.5.5.2 NFC write() function

The write() function is the section which writes the NDEF record to the NFC tag.
To write to the NFC tag the application first creates an NDEF record of the data using the createRecord() function. The createRecord() function creates a record as described in the read() function description. The values are displayed in US-ASCII, as displaying the data

in ASCII format was an easier storage method as opposed to converting the data to hexadecimal and then back to ASCII when the data needs to be read.

Once the record is created for the user entered data the tag is obtained from the intent. Once the tag is identified a connection was made for wireless NFC communication. The stored NDEF data is written to the card using the on-board hardware. The identity number of the patient is stored as the primary key identifier for the data while the rest of the data is stored as the payload.

3.6 PROTOCOL DESIGN

For each method of communication, a protocol is used. In this section each protocol and the design is described.

3.6.1 Protocol Parameters

The parameters used for each protocol are described in table below.

Table 3: Protocol Parameters

Notation	Description
SA	Smartphone Application
DB	Online database and website
PU	User – Patient
MU	User – Medical Practitioner
PR	Personal information record of the user stored on the NFC card
MR	Medical record of the user stored on the medical database includes personal information.
H	Hash function (MD5)
K and K’	Secret key and secondary secret key formed as a function of the original secret key
DE	Data entered by the system user
AG	Access granted flag
T	Generated token containing information set by function
DS	Digital Signature

PV	RSA Private Key
PB	RSA Public Key
AU	Arduino Uno
DT	Disconnected Token
OTP	One-time password
PD	Data stored on the smartphone application accessible only when logged in successfully
PS	Plaintext String
ES	Encrypted String
RS	Random Salt
IV	Initialisation Vector

3.6.2 Patient Application Access (login) Protocol (NFC)

When the login in screen is opened, it has been initiated by the patient. For this protocol the patient with the medical NFC card is the controller.

$$SA \quad : \quad H_{DE}(PU) = f_{DE}(K, DE_{PU})$$

$$SA \quad : \quad f_{DE}(K, DE_{PU}) = H((K' \oplus opad) \,||\, H((K' \oplus ipad) \,||\, DE_{PU}))$$

$$SA \quad : \quad H_{DE}(PU) \leftrightarrow H_{PR}(PU) = AG_1$$

When the user opens the application (SA) they are requested to enter the username and password (DE_{PU}), this data is then hashed using the secret key (K) stored on the NFC card. K' is derived from K on the smartphone through padding K with zeros to the right until the key size is 512 (same as the MD5 block size). Using the standard HMAC RFC 2104 [24] definition DE_{PU} is hashed $H_{DE}(PU)$ and compared to the stored hash value on the NFC tag ($H_{PR}(PU)$). If these values match, then the user first access granted flag is raised (AG_1) on the system.

The next section happens once the AG_1 flag is raised.

$$SA \quad : \quad T_{PR}(PU) = H_{PR}(PU)$$

$$SA \rightarrow DB \quad : \quad T_{PR}(PU) \leftrightarrow DB_{MR}(PU) = AG_2$$

$$DB \rightarrow SA \quad : \quad AG = AG_1 \wedge AG_2$$

The smartphone application sends a generated token containing the user's hashed ID number which is stored on the NFC card to the website containing the database (DB). The database then verifies the token against the data stored on the database and a second access granted flag is raised. Once it has been verified that both the flags are raised, the user is granted access.

3.6.3 Medical Practitioner Patient Portal Access Protocol (NFC)

This section describes the protocol for the medical practitioner to access a patient's medical data, this step comes after the medical practitioner has followed Chapter 3.6.2 to log in to his own portal.

$$SA_{MU} \quad : \quad DS_{PU} = f_{DS}(PV_{PU}, DE_{PU}, PR_{MU})$$
$$SA_{MU} \rightarrow DB \quad : \quad AG = f_{DS}(f_{DS}, PB_{PU})$$
$$DB \rightarrow SA_{PU} \quad : \quad AG(PR_{MU})$$

The medical practitioner's application (SA_{MU}) generates the digital signature (DS_{PU}) using the patients private key (PV_{PU}) stored on the patient's NFC tag. The digital signature is generated from the previously established password (DE_{PU}) and is sent along with the medical practitioner's ID number (PR_{MU}). The access granted flag (AG) is raised when the information is verified on the website using the patient's pubic key (PB_{PU}). Once the access is granted the website sends a notification to the patient's application (SA_{PU}) that the medical practitioner (PR_{MU}) has accessed the database.

3.6.4 Patient Application Access (login) Protocol (Disconnected Token)

The disconnected token access protocol works in a similar way to the NFC card, the protocol is described below.

$$\begin{aligned}
\text{AU} &: \quad H_{PU}(DT) = f_{PU}(OTP) \\
\text{AU} \rightarrow \text{DB} &: \quad f_{PU}(OTP) = H((K' \oplus opad) \,||\, H((K' \oplus ipad) \,||\, OTP)) \\
SA_{PU} &: \quad f_{DE}(K, DE_{PU}) = H((K' \oplus opad) \,||\, H((K' \oplus ipad) \,||\, DE_{PU})) \\
SA_{PU} \rightarrow \text{DB} &: \quad H_{PU}(DT), H_{PU}(DE) \\
\text{DB} &: \quad H_{PU}(DT) \leftrightarrow H_{PU}(DT) = AG_1 \\
\text{DB} &: \quad H_{PU}(DE) \leftrightarrow H_{PU}(MR) = AG_2 \\
\text{DB} \rightarrow SA_{PU} &: \quad AG = AG_1 \wedge AG_2
\end{aligned}$$

On the push of a button by the user the Arduino Uno (AU) generates a 12-digit random number (OTP), hashes the number using a predetermined key ($f_{PU}(OTP)$ and sends the number to the database online ($H_{PU}(DT)$). The smartphone application requests the user to enter the username, predetermined password and random number shown by the Arduino Uno. The application then sends the hashed number and user entered data to the database ($H_{PU}(DE)$). The website confirms that the one-time password matches (AG_1) and that the username and passwords match (AG_2) and sends the access granted flag (AG).

3.6.5 Medical Practitioner Patient Portal Access Protocol (Disconnected Token)

Once the medical practitioner has completed the steps in 3.6.4, and wants to gain access to a patient medical database the following steps are completed.

$$\begin{aligned}
SA_{MU} &: \quad DS_{PU} = f_{DS}(PV_{PD}, PD_{PU}) \\
SA_{MU} \rightarrow \text{DB} &: \quad AG = f_{DS}(f_{DS}, PB_{PU}) \\
\text{DB} \rightarrow SA_{PU} &: \quad AG(PD_{MU})
\end{aligned}$$

The medical practitioner's application (SA_{MU}) generates the digital signature (DS_{PU}) using the patient's private key (PD_{PU}) stored on the application. The digital signature is generated from the previously established password (PD_{PU}) and is sent along with the medical practitioner's ID number (PD_{MU}). The access granted flag (AG) is raised when the

information is verified on the website using the patient's pubic key (PB_{PU}). Once the access is granted the website sent a notification to the patient's application (SA_{PU}) that the medical practitioner (PD_{MU}) has accessed the database.

3.6.6 Data Transfer Protocol

When transmitting new and updated data to the webserver from either the medical practitioner or the patient, the data needs to be transmitted securely so that it is not intercepted along the way. The following steps are followed when data is transmitted which uses the AES standard protocol.

$$\begin{aligned}
&\text{SA} &&: && \text{ES} = f_{\text{PS}}(\text{RS, K, IV}) \\
&\text{SA} &&: && f_{PS}(\text{RS, K, IV})_1 = \text{K}(\text{PS} \oplus \text{IV}) \\
&\text{SA} &&: && f_{PS}(\text{RS, K, IV})_2 = \text{K}(\text{PS} \oplus f_{PS}(\text{RS, K, IV})_1) \\
&\text{SA} &&: && \ldots \\
&\text{SA} &&: && f_{PS}(\text{RS, K, IV})_{1000} = \text{K}(\text{PS} \oplus f_{PS}(\text{RS, K, IV})_{999}) \\
&\text{SA} \rightarrow \text{DB} &&: && \text{ES} = f_{\text{PS}}(\text{RS, IV, 1000}),\ \text{H}_{\text{PU}}(\text{PD, DS})
\end{aligned}$$

The smartphone application creates a random salt (RS), initialises the secret key (K) and the initialisation vector (IV). Using these strings, the cipher ($f_{\text{PS}}(\text{RS, K, IV})$) is initialised and the function outputs the encrypted data (ES) using ~1000 iterations, which is then sent to the webserver (DB) along with the salt, initialisation vector and the number of iterations. The hashed ID number (PD) and digital signature (DS) is also sent to the webserver.

When the data is received on the webserver's end, the data needs to be decrypted and stored. The following steps are completed to achieve this.

$$\begin{aligned}
&\text{DB} &&: && \text{H}_{\text{PU}}(\text{PD}) \leftrightarrow \text{H}_{\text{PU}}(\text{MR}) = \text{AG}_1 \\
&\text{DB} &&: && \text{H}_{\text{PU}}(\text{DS}) \leftrightarrow \text{H}_{\text{PU}}(\text{MR}) = \text{AG}_2 \\
&\text{DB} &&: && \text{AG} = \text{AG}_1 \wedge \text{AG}_2 \\
&\text{DB} &&: && \text{PS} = f_{\text{ES}}(\text{RS, K, IV})
\end{aligned}$$

DB : $f_{ES}(\text{RS, K, IV})_{1000} = (\text{K(ES)}) \oplus \text{IV}$

DB : $f_{PS}(\text{RS, K, IV})_{999} = (\text{K(ES)}) \oplus \text{ES}$

DB : ...

DB : $f_{PS}(\text{RS, K, IV})_{1} = (\text{K(ES)}) \oplus \text{ES}$

The webserver (DB) verifies the client digital signature (DS) and the ID number (PD) with the values stored on the database (H_{PU}(MR)). Once this is verified successfully (AG) the decryption begins. The encrypted text (ES) is sent through the decryption function the same amount of times and the encryption which is sent by the smartphone application using the initialisation vector, the random salt and the stored secret key. Once the data is decrypted ($f_{PS}(\text{RS, K, IV})_1$) the data is stored in the database under the patient's details.

3.7 CONCLUSION

Chapter 3 covers the architecture and implementation of the patient-based monitoring system.

The transfer of medical information between the devices and the smartphone is completed through the use of Bluetooth and an NFC card.

The protocol is designed to take into account the security of the medical information, medical legislature and POPI. This is completed through the levels of security and authentication to access patient information.

Both AES and Public Key encryption is implemented in various sections of the application to ensure optimum security.

The design is finalised and the tests and results described in Chapter 4.

CHAPTER 4 RESULTS

4.1 CHAPTER OVERVIEW

This chapter covers the testing procedure for the architecture implementations described in Chapter 3. The dissertation utilises technologies and programming practices which have varying transfer speeds. The sectors that are tested are the NFC component, the disconnected token component, the Bluetooth components, the Webserver, the encryption as well as the compression.

The testing procedure for each sector is in detail in the sections below and a complete discussion of the results is completed in Chapter 5.

4.2 SUMMARY OF RESULTS ACHIEVED

In Table 4 below, the results achieved are shown against the required outcomes for the system.

Section	**Experiment**	**Outcome**
NFC	Data throughput to smartphone from NFC tag	0.119 Mbps
	Data throughput to NFC tag from smartphone	0.1 Mbps
Disconnected Token	Data throughput to webserver from disconnected token	4.63 Mbps
Bluetooth	Data throughput to smartphone from Bluetooth Thermometer	0.342 Mbps
	Data throughput to smartphone from Bluetooth Blood Glucose Monitor	0.393 Mbps
	Data throughput to smartphone from Bluetooth Blood Pressure Monitor	0.182 Mbps
	Arduino Uno and Bluetooth Bee data throughput to smartphone	0.996 Mbps
Webserver	Data throughput to online database from smartphone	3.61Mbps

	Data transfer reliability to online database from smartphone	100%
Encryption (AES)	Encryption Speed	78 ms
	Decryption speed	84 ms
	Data reliability once decrypted	100%
Encryption (Public Key - RSA)	Signing Speed	52 ms
	Verifying speed	8 ms
	Data reliability once decrypted	100%
Complete System using NFC tag	Complete system time recorded when the NFC tag is used	1216 ms
Complete System using disconnected token	Complete system time recorded when the disconnected token is used	4129 ms

Table 4

Summary of the results achieved.

4.3 QUALIFICATION TESTS

4.3.1 Component test plans and procedures

Component Test Strategy Overview

The testing strategies for the components involved in the system are described in this section. The NFC and Bluetooth communication throughput and latency were tested and in addition, the AES and Public Key cryptography are separately tested. Each testing procedure is described in the following sections. The results will be discussed in Chapter 5.

4.3.2 Experiment 1: NFC Transfer Speed Test Procedure

4.3.2.1 Objectives of the experiment

This experiment determines the throughput and latency when transferring a file from the smartphone application to the NFC card.

4.3.2.2 Component Test Plan and Procedure

For the first test of the data throughput a 1 KB file is transferred through the application to the smartphone, the second test reads the 1 KB file from the NFC tag to the smartphone. For the first test the amount of time for the application to send the file over and state that the file had been successfully transferred is recorded. The second test measures the amount of time for the read request to be sent and the complete file to be transferred. Using this data, the throughput and latency are calculated.

In this test and certain tests further on the time calculated for each data transfer cycle is completed through the use of the mobile application or the webserver. When the medical devices or the NFC tag is used, the mobile application runs a procedure to calculate the time for transfer. When the user presses the transfer button the following line of code is run to determine the time in milliseconds:

```
long currentT = System.nanoTime();
```

Once the transfer is completed successfully by the smartphone, the following line of code is run to determine the total time taken to complete the task.

```
long timeTaken = System.nanoTime() - currentT;
```

For this test a Samsung Galaxy A5 smartphone running the application on Android 4.2 and a Trikker 2K NFC tag are used.

4.3.2.3 Results and observations

The results achieved for each run of the experiment are shown in Figures 16 and 17. The experiment was run 30 times in total.

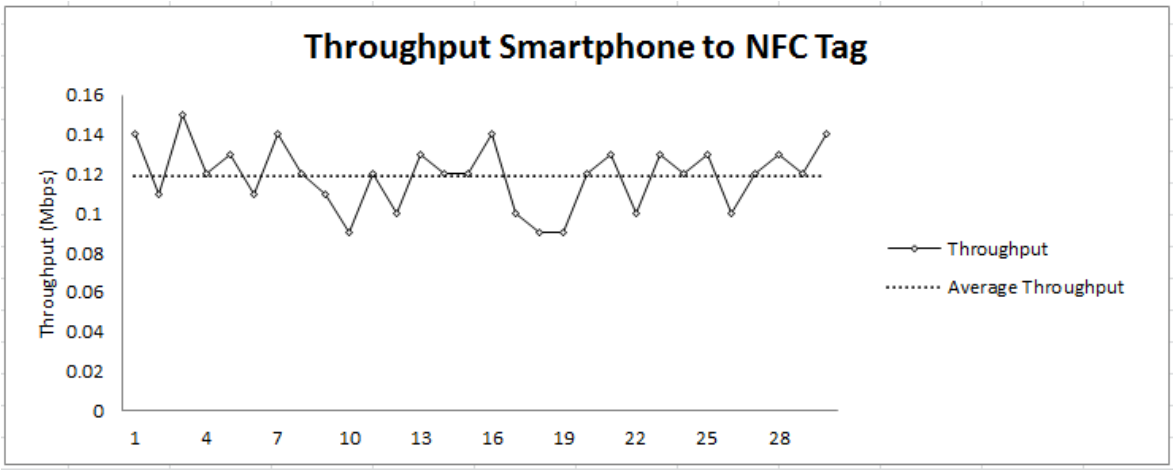

Figure 16: Line graph of the throughput from the smartphone to the NFC tag

The variance for the throughput was determined to be 0.000264 and the standard deviation 0.016226.

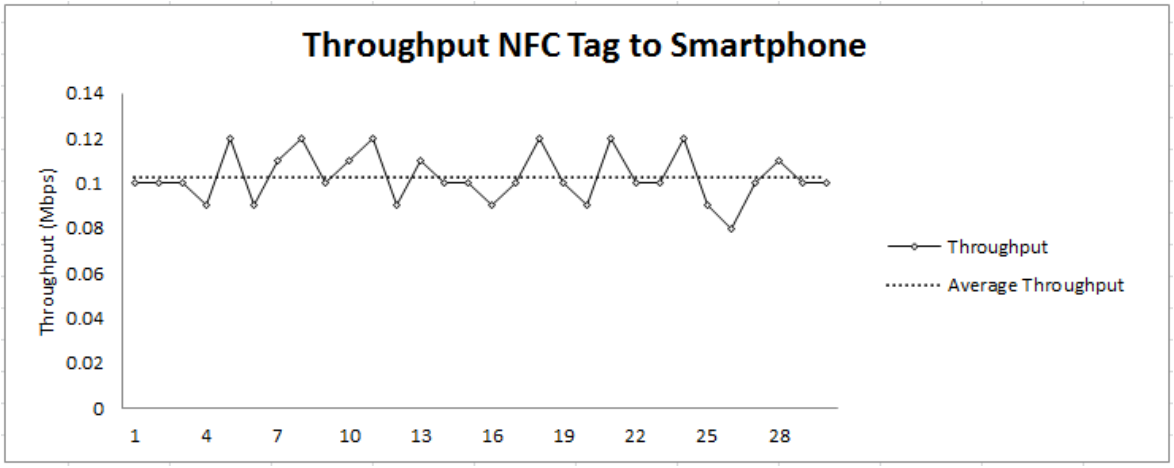

Figure 17: Line graph of the throughput from the NFC tag to the smartphone

The variance for the throughput was determined to be 0.000124 and the standard deviation 0.011121.

4.3.3 Experiment 2: Disconnected Token Transfer Speed Test Procedure

4.3.3.1 Objectives of the experiment

This experiment determines the throughput when transferring a file from the Arduino Uno to the webserver.

4.3.3.2 Component Test Plan and Procedure

To test the transfer speed of the Arduino Uno and Wi-Fi Shield to the webserver a test file of 1 KB is used. The test file contained false data for experimental purposes. The amount of time for the file to be transferred from the Uno to the webserver is measured and the throughput calculated from the data.

An Arduino Uno with a Wi-Fi attachment was used for this test on a 6 Mb Telkom line. The database is loaded on a webserver hosting site.

4.3.3.3 Results and observations

The results achieved for each run of the experiment are shown in Figure 18. The experiment was run 30 times in total.

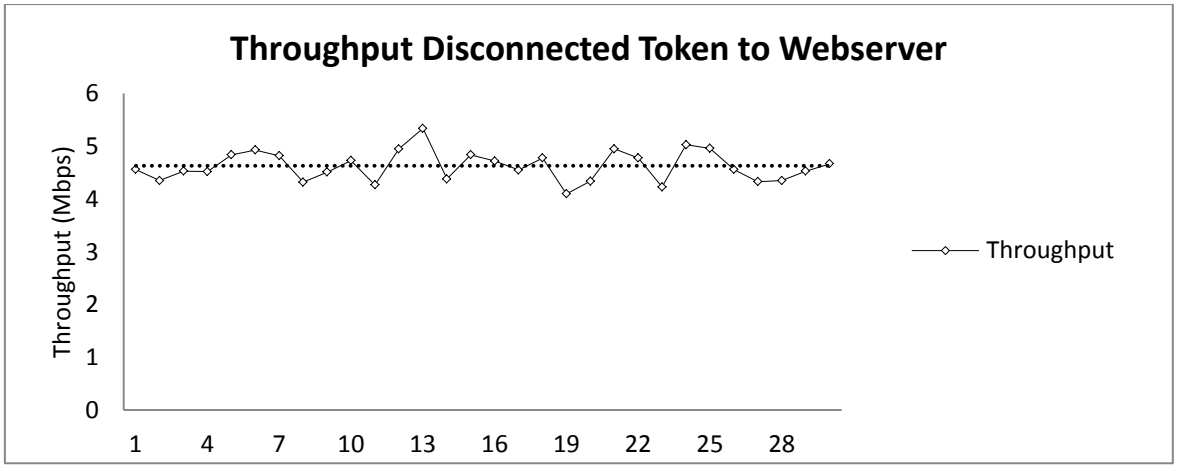

Figure 18: Line graph of the throughput from the Disconnected Token to the Webserver

The variance for the throughput was determined to be 0.082 and the standard deviation 0.2864.

4.3.4 Experiment 3: Bluetooth Transfer Speed Test Procedure – Medical Devices

4.3.4.1 Objectives of the experiment

This experiment determines the throughput and latency when transferring a file from the Bluetooth medical devices to the smartphone application.

4.3.4.2 Component Test Plan and Procedure

Each medical device, the blood glucose monitor, the blood pressure monitor and the temperature monitor are tested separately. Each device is run 30 times and the readings transferred to the smartphone application via the Bluetooth interface. The speed of transfer once the vital sign reading was complete is measured. To calculate the time taken for the medical devices to transfer the medical data to the smartphone; the time for the connection to the Bluetooth device is recorded using:

```
long currentT = System.nanoTime();
```

And once the transfer of the medical information is complete and successful the second line of code runs:

```
long timeTaken = System.nanoTime() - currentT;
```

The Fora IR20 ear thermometer, Fora P20b Blood Pressure monitor, Fora G31 Blood Glucose monitor as well as a Samsung Galaxy A5 smartphone running the application on Android 4.2 are used for this test.

4.3.4.3 Results and observations

The results achieved for each run of the experiment are shown in Figures 19, 20 and 21. The experiment was run 30 times in total for each device.

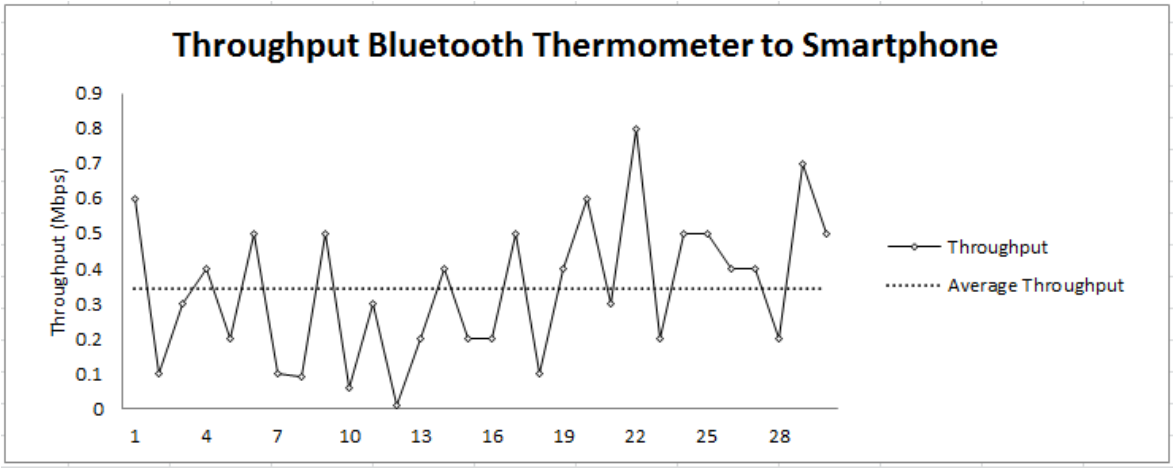

Figure 19: Line graph of the throughput from the Bluetooth Thermometer to the Smartphone

The variance for the throughput was determined to be 0.041 and the standard deviation 0.2028.

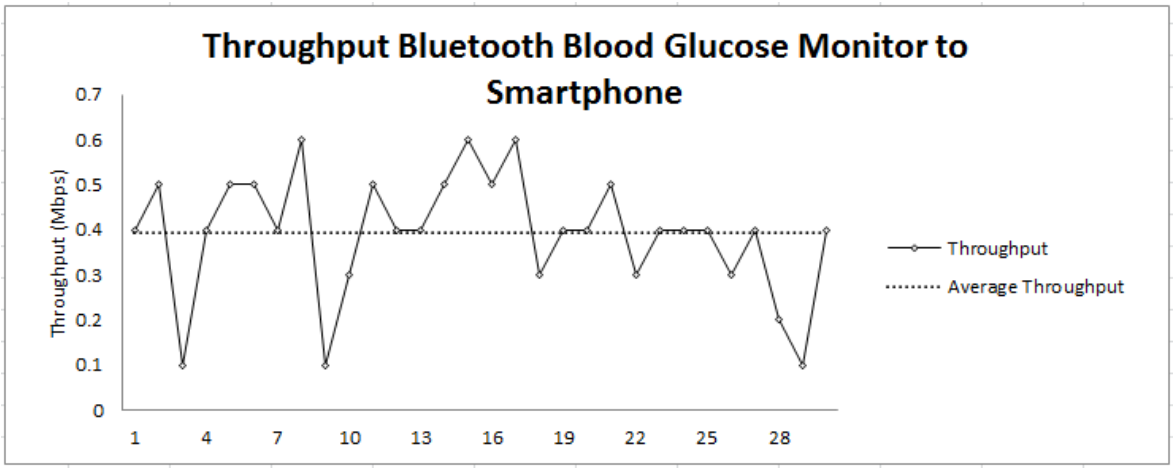

Figure 20: Line graph of the throughput from the Bluetooth Blood Glucose Monitor to the Smartphone

The variance for the throughput was determined to be 0.0186 and the standard deviation 0.136.

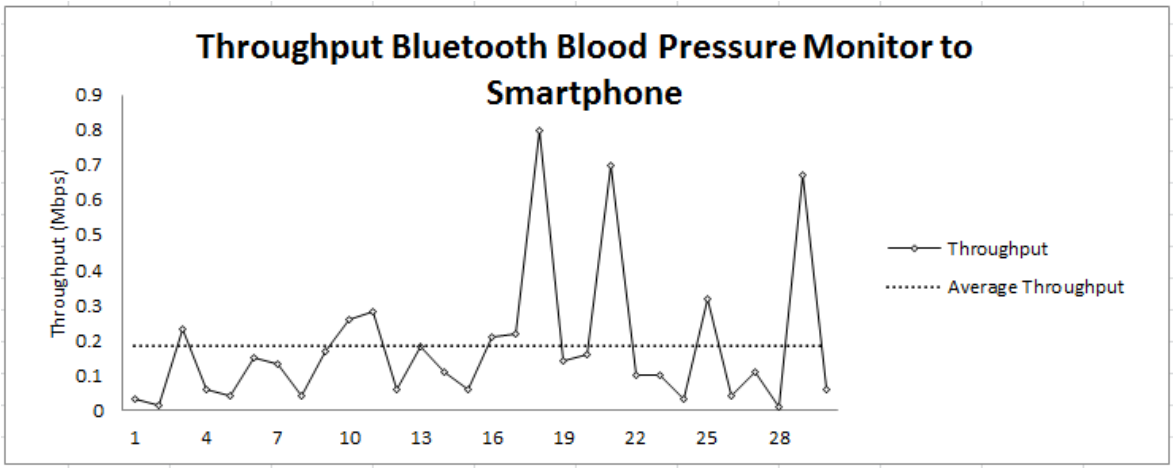

Figure 21: Line graph of the throughput from the Bluetooth Blood Pressure Monitor to the Smartphone

The variance for the throughput was determined to be 0.0408 and the standard deviation 0.2018.

4.3.5 Experiment 4: Bluetooth Transfer Speed Test Procedure – Arduino Uno and Bluetooth Bee

4.3.5.1 Objectives of the experiment

This experiment determines the throughput and latency when transferring a file from the Arduino Uno and Bluetooth Bee to the smartphone application.

4.3.5.2 Component Test Plan and Procedure

The Arduino Uno is set up to send blood pressure readings to the application. The Bluetooth Bee sends 30 readings to the application waiting 3 minutes between transfers to simulate the time taken for the medical device to take the reading. The 3 minutes are not included in the speed calculation. The speed of transfer once the reading was complete is measured. To calculate the time taken for the medical devices to transfer the medical data to the smartphone; the time for the connection to the Bluetooth device is recorded using:

```
long currentT = System.nanoTime();
```

And once the transfer of the medical information is complete and successful the second line of code runs:

```
long timeTaken = System.nanoTime() - currentT;
```

The Arduino Uno with a Bluetooth Bee attached as well as a Samsung Galaxy A5 smartphone running the application on Android 4.2 are used for this test.

4.3.5.3 Results and observations

The results achieved for each run of the experiment are shown in Figure 22. The experiment was run 30 times in total.

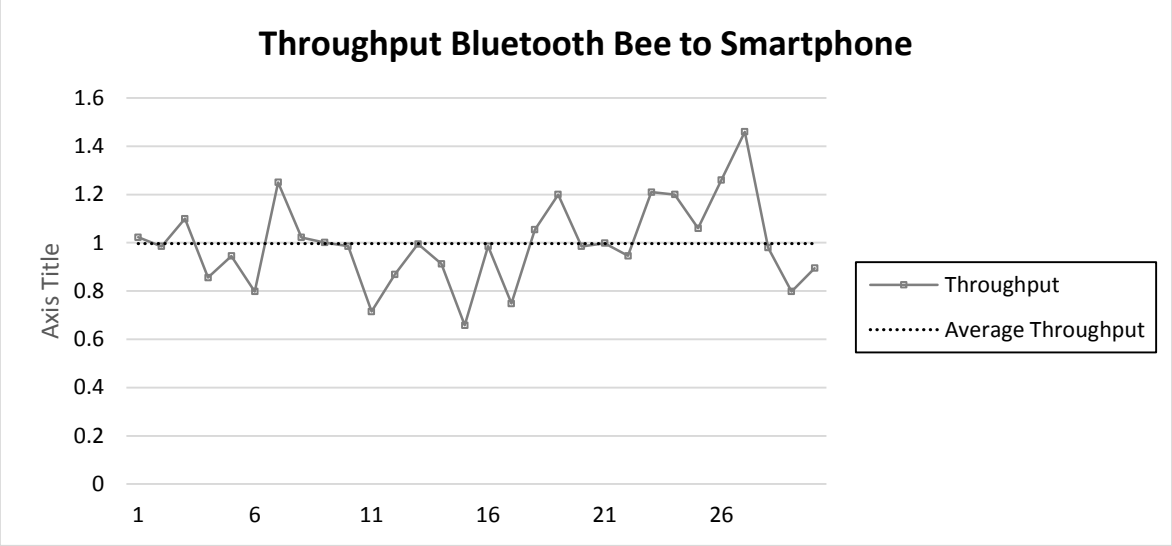

Figure 22: Line graph of the throughput from the Bluetooth Bee to the Smartphone

4.3.6 Experiment 5: Webserver Transfer Speed Test Procedure

4.3.6.1 Objectives of the experiment

This experiment determines the throughput and latency when transferring a file from the smartphone application to the online database.

4.3.6.2 Component Test Plan and Procedure

To test the transfer to the medical database from the smartphone, a file with medical data including readings from the medical devices is sent to the database along with the identifying primary key for the database (ID number). In addition, the time recorded by the smartphone to the millisecond is sent to the database. The amount of time for the database

to receive, store and verify the successful transfer is measured through a verification flag being set to true when the transfer is complete. When the flag is set to true, the current time to the millisecond is recorded and subtracted from the time sent from the smartphone. This is the time recorded as the complete smartphone to webserver transfer.

A Samsung Galaxy A5 smartphone running the application on Android 4.2 are used for this test on a 6 Mb Telkom line. The database is loaded on a webserver hosting site.

4.3.6.3 Results and observations

The results achieved for each run of the experiment are shown in Figure 23. The experiment was run 30 times in total.

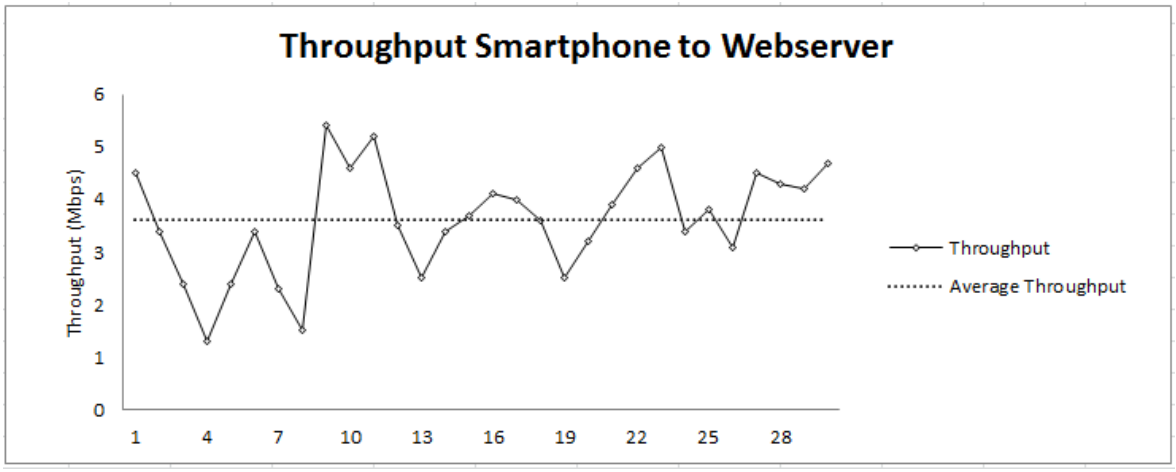

Figure 23: Line graph of the throughput from the Smartphone to the Webserver

The variance for the throughput was determined to be 1.078 and the standard deviation 1.038.

4.3.7 Experiment 6: Total System Time Test Procedure

4.3.7.1 Objectives of the experiment

This experiment determines the total time taken to complete the system process from login to storage on the database.

4.3.7.2 Component Test Plan and Procedure

To test the complete time taken for the transfer 3 users who are familiar with the system (Users 1-3) and 3 users who are not familiar with the system (Users 4-6) were requested to run the application from login to data storage on the webserver. This was done to ensure that the system is user friendly to both experienced users and inexperienced users, and that the time taken for transfer is not adversely affected.

The system was then run with the users recording their vital signs using the thermometer. When the user opens the application the current time is recorded, this is then subtracted from the time when the login screen grants the user access. The users are then requested to use the thermometer. Once the recording is completed the Bluetooth connection is made to the smartphone application and the current time is recorded, this is used as the start of the transfer time. The vital sign is then sent to the smartphone, which forwards the data to the webserver along with the patient's information. The webserver then verifies and stores the data. Once the data has been stored successfully the current time is recorded and subtracted from the time recorded at the medical device's Bluetooth connection. The total time taken is the login time (using either the NFC card or the disconnected token) added to the processing and storing time.

A Samsung Galaxy A5 smartphone running the application on Android 4.2 are used for this test on a 6 Mb Telkom line. The database is loaded on a webserver hosting site. The token being used is the Arduino Uno with Wi-Fi module or the NFC medical card.

4.3.7.3 Results and observations

The results achieved for each run of the experiment are shown in Figure 24. The experiment was run 120 times in total; each user tested the system 10 times using the NFC token, and 10 times using the Disconnected Token (Figure 25).

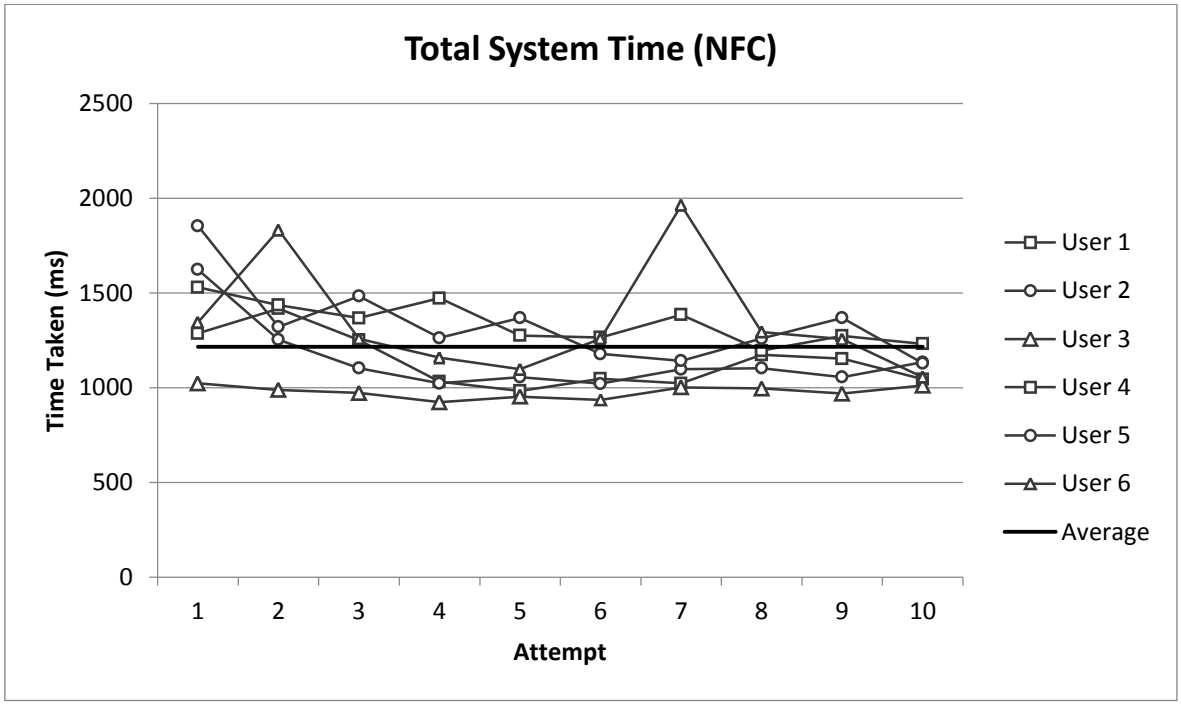

Figure 24: Line graph of the average and user system time for the NFC

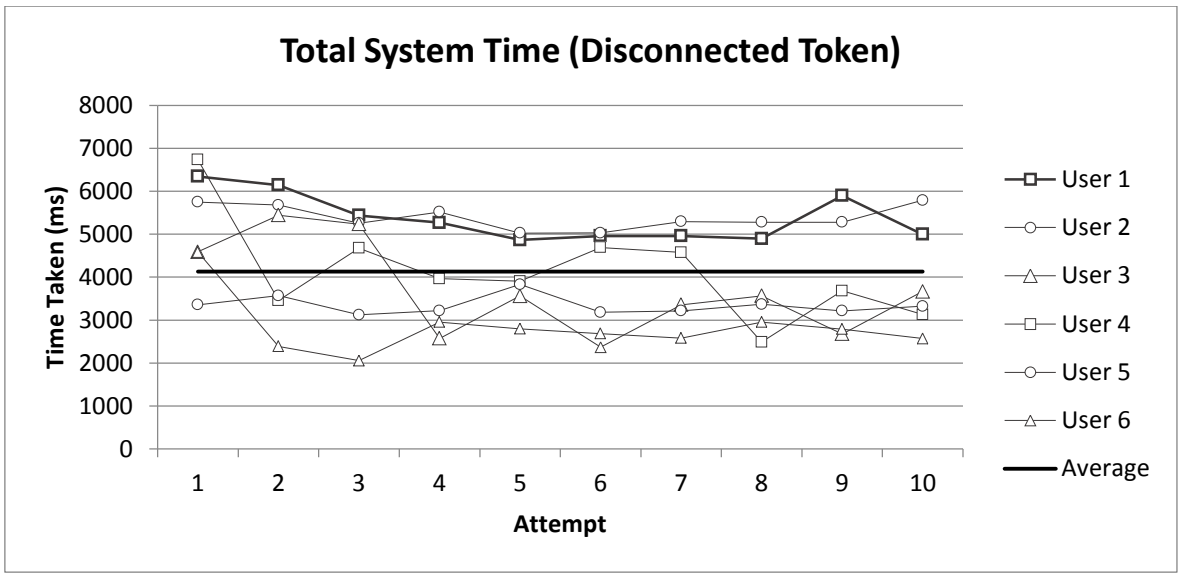

Figure 25: Line graph of the average and user system time for the Disconnected Token

4.4 CONCLUSION

The tests for each component of the smartphone application, data transfer and security are tested in this chapter.

Each test is described in detail with the test conditions, the component test plan and the objectives of the experiment. The results are discussed in Chapter 5.

CHAPTER 5 DISCUSSION

5.1 CHAPTER OVERVIEW

This chapter covers the discussion of the experiments performed in Chapter 4 as well as a complete discussion on the overall system and the research completed using the system. The first section is a discussion of the test results and considerations. The next section is a complete discussion on how the design complies with the legislation discussed in Chapter 2.

5.2 DISCUSSION OF TEST RESULTS

5.2.1 NFC transfer speed

The NFC medical card is the item which every person on the global medical database needs to have. The card is treated as another form of identity in that it should be carried on the person at all times. For this reason, the card cannot be a heavier or unwieldy form of communication. In addition, the communication needs to be almost instantaneous as a slower method of communication caused impatience or critical delays when dealing with emergencies.

The NFC tag transfers the 1 kB file to the smartphone at an average speed of 0.12 Mbps. The variance for the throughput is determined to be 0.000264 and the standard deviation 0.01623.

The data transferred from the NFC tag consists of information such as encrypted or hashed personal information, collected through an NDEF message using the application's read() method. It can be determined from this information that the information sent to the application from the NFC card never exceeded 1kB. At a speed of 0.12 Mbps the time taken for this information transferral is 0.12s making it imperceptible to the user.

The transfer speed of information from the application to the NFC tag is 0.1 Mbps. This speed is also imperceptible to a casual user. The variance for the throughput was determined to be 0.000124 and the standard deviation 0.011121.

The data transferred to the NFC tag contains more data than the transfer from the NFC tag, however that data is only transferred when the user first creates the card information and then updates the information on the card. The information consists of all the patients' medical data, as well as the digital signature data.

5.2.2 Disconnected token transfer speed

The disconnected token is treated the same way as the NFC medical card. Every person on the medical database is issued a token. The token is required to be carried on the person at all times, however considering that the token can be created as a key fob, the token cannot have as much data printed on it as a NFC medical card.

The Arduino Uno Wi-Fi shield transfers the 1KB file at a speed of 4.63 Mbps. The variance for the throughput was determined to be 0.082 and the standard deviation 0.2864.

The data transferred by the disconnected token is only the hash value of the token generated, however it can be modified to store more information if needed. It was however determined that the amount of data transferred by the disconnected token is less than 1KB as anything larger negates the use of the database.

The disconnected token has a few drawbacks, the device needs to be charged regularly, which may not be possible in rural areas of South Africa and the device is also more susceptible to damage when compared to the NFC card which is smaller and is stored in a wallet.

5.2.3 Bluetooth transfer speed

The Fora blood pressure monitor, blood glucose monitor and ear thermometer used in this dissertation performs as expected when working with Bluetooth transfer speeds. The average data transfer speed of the Blood pressure monitor is 0.182 Mbps, which to an average user of the application is imperceptible, this speed allows for the application to receive accurate data in a matter of seconds. The variance for the throughput is determined to be 0.0408 and the standard deviation 0.2018.

In addition, the blood glucose monitor's average data transfer speed is recorded to be 0.393 Mbps, with the variance for the throughput determined to be 0.0186 and the standard deviation 0.136.

While the ear thermometer's data transfer speed is recorded to be 0.342 which is also imperceptible to the average user. The variance for the throughput is determined to be 0.041 and the standard deviation 0.2028.

When comparing these numbers to the average data transfer sped of the Arduino Uno and the Bluetooth Bee average transfer speed at 0.996, it can be seen that the speed of transfer is not significantly slower in the medical devices.

There were certain users who excelled with certain tests; these outliers can be explained by the user being more confident with the use of the application after multiple tries. When the user lagged behind the average at certain tests it was noted the user attempted to rush through the test, and began to make errors or going to the incorrect sections within the application for the purposes needed at the time.

The medical devices used for this dissertation have built in Bluetooth transfer protocols. This creates the need for the smartphone application to be optimised to the protocol used in the devices. For this dissertation, the Bluetooth implementation and transfer speed of devices such as the Fora devices are compared to the implementation and transfer speed of a Bluetooth device created for the sole reason of the application.

The implementation of the coding of the Arduino Uno and Bluetooth Bee compared to the implementation of the coding for each medical device's Bluetooth protocol is negligible, however the ease of creating one Bluetooth device which can read multiple medical devices rather than multiple medical devices with a separate Bluetooth module is a greater advantage.

Creating a standalone Bluetooth device can be highly beneficial in the rural areas where it is cheaper to buy a simple medical device and make the device compatible with a single universal Bluetooth device.

5.2.4 Webserver transfer speed

Sending, securing and verifying the data on the webserver is the most important part of the application. This section ensures that the patient data is located in a secure database, stored under the correct patient name and contained the most up to date data taken from the patient.

For this section the speed of transfer averaged 3.61 Mbps, which means that when sending patient data to the database, there is no humanly perceptible delay. If a doctor is viewing the transferred data offsite, the data will be available immediately and the doctor can use the data as applicable.

As the data sent to the database is critical patient information the accuracy cannot be low. When testing the data read from the medical devices and the data sent by the application to the webserver and comparing it to the data recorded and stored on the webserver, there were no variations i.e. 100% accuracy. This allows medical practitioners to put their full trust in the system, as opposed to handwritten data which can be compromised by human error, which can contain up to 23% of data missing or unreadable [11].

5.2.5 AES and RSA encryption

When testing the encryption and decryption speed of AES and the signing and verifying speed of RSA it was determined that RSA is slightly faster. However with increases data size and complexity, RSA becomes significantly slower than AES. For this application it was decided that both encryption standards will be used.

As the difference in speed between AES and RSA is negligible, it was decided that AES will be used to store and transmit the user's data securely. This was due to the decrease in performance in RSA when larger file sizes were encrypted.

Although RSA significantly slows down when the data is larger it will be used for the digital signature, key transmission and verification of permissions as these consist of smaller sized data which needs to be transmitted faster.

5.2.6 Complete system speed and usability

The autonomous transfer of a file containing dummy medical patient data from the login screen to data capture on the database using the NFC tag typically took between 1.324s to 2.845s. These values exclude any human input to the system.

Using the same experiment with the disconnected token took between 12s and 13s; however the insertion of the 12-digit code was done by a single experienced user.

The average transfer time recorded for both experienced and inexperienced users to use the system was 12s from logon to data capture on the webserver using the NFC tag. This time excluded the time taken to get a reading from the patient on the medical device.

The average transfer time recorded for both experienced and inexperienced users to use the system is 41s from logon to data capture on the webserver using the disconnected token.

The reason the disconnected token took a much longer time for the users to login was that manually typing the 12-digit code caused multiple problems. Quite often incorrect digits were entered and the application did not log the user in. In addition transferring the digits from the screen to the application was difficult for certain users who were not comfortable with switching between capital and small letters, numbers and characters on the smartphone.

It can be seen through the experiment that the system runs efficiently and effectively when users are not included in the calculation. The delay in the complete system run time was substantially increased when the user ran the system. However, this time delay can be reduced once all users get more comfortable with using the system on a regular basis. The time to run the complete system without user input is not a deterrent to the users as there is no perceptible delay due to the system.

Although the NFC card had a slower transfer speed, the overall system was faster using the NFC card when compared to the disconnected token.

When a manual method of recording was used to log data onto the system, it took multiple hours; as there were many considerations. The medical health practitioner needed to record the medical data onto a file. Multiple other patient recordings may be taken before the patient's file has been handed in for logging on the database. If the practitioner is in a rural area with no access to the computer or database, the file will only be handed in for logging

onto the database when the practitioner returns to the office. The practitioner or aide will then need to log the data manually onto a computer or online database.

When comparing the application to a manual system the application proved to be much more efficient.

5.3 DISCUSSION OF MEDICAL LEGISLATION ADHERENCE

In Chapter 2.5 the legislation for keeping of medical health records in South Africa is described. To determine if this application and database implementation is allowed to operate in South Africa, it has to be determined whether all the rules have been followed.

1. The application may not allow any information to be removed or altered from the medical health record.

When the database has information added to it, no information was altered or removed, rather the information is appended to the end of the health record with a time stamp. The database is only accessible through the application to medical practitioners and patients, the only data which can be altered is the personal information of the patient, this information can only be altered by the patient him- or herself. When an alteration was made, the previous entry is altered to have a strikethrough. The information is still visible and accessible; however, the new information is prominent.

2. An error must be corrected by placing a line through it and with ink and correcting it. And any errors or amendments must have a valid reason for the change attached to the error.

Since the information recorded by the application is only done through medical health devices with no human interaction, the data will not need to be corrected. If there is an error with the medical data reading, the doctor retakes the test and labels it as a second reading at the same time with an attached reason.

3. Additional entries must be dated and signed

All medical health records and readings are time-stamped.

4. Health records should be stored for a period of not less than 6 years from the date they become inactive or dormant.

The medical database contains all entries of the patient since the NFC card was issued. To comply with the legislation, the medical data is archived after 6 years of inactivity; however, this data will be stored in a secure secondary database indefinitely.
The NFC medical card will be issued to new-born babies' parents on the same date as the birth certificate is issued. The parents will be in control of the card until the child is old enough.
According to the Children's Act No 38 of 2005 a child 12 years or older may have access to his or her own records, however this must be provided by a practitioner. In this case, the practitioner may apply for the child to have full control of the NFC card with their own password. Once the child has reached the age of 16, the NFC card will be transferred to the child's name and they will be allowed to alter the password.

The South African National Health Act outlines the protection of medical data records by using the following guidelines [28]:

1. The person in charge of the health establishment which has possession of any health records must have control measures in place which prevent the unauthorised access to the medical records and the storage facility in which they are kept.
2. Records must also be protected from damage, in particular digital records should be backed up regularly and the backup should be kept in a separate off-site location.

For this dissertation to comply with the Act, the database only allows monitored and permissioned access to the database. In addition, the database was set to back the data up automatically on a regular basis. The current database was simulated on a 3rd party database however, for the implementation of this dissertation on a nationwide scale, the database will be controlled by the government and the backups will be stored off site and in multiple copies.

5.4 SYSTEM COMPARISON

In Chapter 2.6 currently implemented similar medical systems are discussed. This section compares the systems to the system proposed in the dissertation.

5.4.1 Comparison Table

Table 5 below briefly compares the devices discussed in Chapter 2.6; a full description of each section is in Chapter 5.4.2 and 5.4.3.

Table 5: Medical Device Comparison

	Medical Health Sensor Device	Health Information Exchange	mHealth
Interoperability	✓	✓	×
Comprehensiveness	✓	✓	✓
Legal Value	✓	✓	×
Availability	✓	✓	✓
Open Source code	✓	×	×
Security	✓	×	×

5.4.2 Health Information exchange

When discussing the HIE in Chapter 2 the design considerations are listed. To adequately compare the system designed in this dissertation to the HIE the design considerations are evaluated with regard to the dissertation.

1. Interoperability – The system designed in this dissertation takes interoperability into consideration as the protocols can be adapted to many medical applications as the design was catered to a generic system.
2. Comprehensiveness – The medical records stored in the database contains all the medical information needed by the South African government as outlined by the legislation mentioned in Chapter 5.3.
3. Legal Value – The system designed in this dissertation was catered to adhere to the laws in South Africa. The patient has full access to the heath records and can alter

any personal information. The patient may also grant access of the information to the medical health practitioner of their choice. The Patient is notified when the medical database is accessed and by who.

4. Availability – Medical health practitioners have access to the information when provided by the patient as well as emergency access when the patient is not able to provide access.

5.4.3 mHealth

The mHealth and MomConnect system is a currently successful implementation of a medical health application which sends out eHealth information and shares medical data between facilities. The challenges encountered by the system were identified. The system designed in this dissertation is able to provide a solution of most of these challenges as listed below.

1. Lack of interoperability – The protocols designed in this dissertation will allow for interoperability when implemented across all applicable systems.
2. Absence of a single framework – This problem is not addressed in this dissertation.
3. Lack of use of open-source options – Android is an open source coding language and the code for the application can be made available.
4. Absence of practical approaches to privacy and security – the protocols designed in this dissertation takes into account all foreseen privacy issues as the patient is notified of any access to the database and can easily view any changes. The security designed in this solution also adheres to POPI and allows for the patient's data to be secure.

5.5 CONCLUSION

This chapter covers the discussion around the results achieved in Chapter 4. The transfer speed of each technology tested is compared and the adequate solution for the system was found. It has been decided that the NFC token was the best solution and that the transfer speeds for the Webserver and Bluetooth were adequate for the applications needed.

The encryption speed and implementation were tested and both methods, AES and RSA are to be used. However RSA will be used when smaller bits of data needed to be transferred, such as the digital signature. Two methods of encryption allows for a more secure system.

The adherence to South African Legislation and the Personal Protection of Information Act was discussed and the conclusion drawn that the system is compliant.

The system compared well to currently adopted medical health systems; it was found that the considerations and challenges found in both the mHealth system and the Health Information Exchange system are addressed.

CHAPTER 6 CONCLUSION

The medical health field is moving toward more technology centred applications; however, in South Africa, this move has been slower due to multiple factors such as the extent of rural areas in South Africa and the high cost of technology.

The recording of medical data is done by the medical practitioner and it has been shown that this data can be erroneous. The application and database presented in this dissertation addresses this problem while adhering to the legislation and laws around medical record keeping.

The design and performance of the application and data transfer is completed and it was found that the system is a viable option urban South Africa; however test conditions within rural South Africa will need to be considered in future. The medical data recording is completed digitally, the patient information is stored securely and the information is accessible to both the patient and the medical practitioner through the use of a smartphone.

The application allows a system to be created whereby South African citizens will be issued a medical health NFC card continuing access to a database where all the medical information is stored. The NFC card is protected using MFA.

It ensures that medical information is only accessible with the patient's permission and that the patient is informed of access to their information.

The system allows for redundant tests to be eliminated as all information is stored on one global database. If a doctor requires a test to be performed which was otherwise done by another professional, the results of that test are accessible to the current health practitioner to ensure that the patient is not subjected to the test unless completely necessary.

This dissertation also showed that a secure system is possible without long time delays for the medical practitioner. It ensures that recording the data through the application was faster that recording by hand so that practitioners are prevented from using the system due to delays.

When compared to similar systems within South African it was found that the system in this dissertation addressed many of the considerations and challenges found in the other systems.

6.1.1 Future considerations

When completing the literature study and deciding on the most optimal design to use for this dissertation the system is not discussed with medical health practitioners. In future, the system should be tested by the medical health practitioners over a long time period to determine any advantages, suggestions for improvement, flaws and disadvantages found in the system. The system may not account for problems which medical health practitioners have found working in the field.

In this system using methods other than Bluetooth to collect information from medical health devices (such as Wi-Fi) is excluded due to the lack of universal Wi-Fi in South Africa. However, if the system were to be implemented in other countries, Wi-Fi can be a viable option.

REFERENCES

[1] S. Buhlungu, J. Daniel, and R. Southall, “Public Hospitals in South Africa: Stressed Institutions, Disempowered Management,” State of the Nation: South Africa 2007, vol. 1, no. 1, pp. 312–314, 2007.

[2] M. Srivastava, M. Hansen, J Burke, A. Parker, S. Reddy, G. Saurabh and D. Estrin, “Wireless urban sensing systems,” Center for Embedded Network Sensing, pp 53-57, 2006.

[3] Stankovic, J. A., Q. Cao, T. Doan, L. Fang, Z. He, R. Kiran, S. Lin, S. Son, R. Stoleru, and A. Wood, "Wireless sensor networks for in-home healthcare: Potential and challenges", *High confidence medical device software and systems (HCMDSS) workshop*, pp. 7-10, 2005.

[4] P. Adrian, J. A. Stankovic, and D. Wagner, “Security in wireless sensor networks,” *IEEE Trans. Inf. Technol.*, vol. 47, no. 6, pp. 53–57, 2004.

[5] J. Stankovic, T. Abdelzaher, Chengyang Lu, Lui Sha and J. Hou, "Real-time communication and coordination in embedded sensor networks", *Proceedings of the IEEE*, vol. 91, no. 7, pp. 1002-1022, 2003.

[6] A. Wood and J. Stankovic, "Denial of service in sensor networks", *Computer*, vol. 35, no. 10, pp. 54-62, 2002.

[7] T. Abdelzaher, Tian He and J. Stankovic, "Feedback control of data aggregation in sensor networks", *43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, vol. 2, pp. 1490-1495, IEEE, 2004.

[8] R. Stoleru and J. Stankovic, "Probability grid: a location estimation scheme for wireless sensor networks", *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004.* pp. 430-438, IEEE, 2004.

[9] M. Lange, S. Liebergeld, A. Lackorzynski, A. Warg and M. Peter, "L4Android : a generic operating system framework for secure smartphones.", *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices - SPSM '11*, pp. 39-50, ACM, 2011 .

[10] H. Chon S. Jun, H. Jung and W. An, "Using RFID for Accurate Positioning", *Journal of Global Positioning Systems*, vol. 3, no. 1 & 2, pp. 32-39, 2004.

[11] M. K. E. B. Wallin and S. Wajntraub, “Evaluation of Bluetooth as a Replacement for Cables in Intensive Care and Surgery,” *Anaesthesia & Analgesia*, pp. 763–767, 2004.

[12] K. Hung, Y. T. Zhang, and B. Tai, “Wearable medical devices for tele-home healthcare,” *Proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. vol. 7, pp. 5384–5387, 2004.

[13] K. Puttaswamy, C. Kruegel and B. Zhao, " Silverline: toward data confidentiality in storage-intensive cloud applications. ", *Proceedings of the 2nd ACM Symposium on Cloud Computing - SOCC '11*, p. 10, ACM, 2011.

[14] G. Kambourakis, E. Klaoudatou and S. Gritzalis, "Securing Medical Sensor Environments: The CodeBlue Framework Case", *The Second International Conference on Availability, Reliability and Security (ARES'07)*, pp. 637-643. IEEE, 2007.

[15] L. Ruiz, J. Nogueira and A. Loureiro, "MANNA: a management architecture for wireless sensor networks", *IEEE Commun. Mag.*, vol. 41, no. 2, pp. 116-125, 2003.

[16] E. Oyman and C. Ersoy, "Multiple sink network design problem in large scale wireless sensor networks", *2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577)*, vol. 6, pp. 3663-3667. IEEE, 2004.

[17] M. Pasquet, J. Reynaud, and C. Rosenberger, “Secure Payment with NFC Mobile Phone in the SmartTouch Project,” *Collaborative Technologies and Systems 2008. CTS 2008. International Symposium*, pp. 121–126, 2008.

[18] S. Bajaj and R. Sion, “TrustedDB: A Trusted Hardware-Based Database with Privacy and Data Confidentiality,” vol. 26, no. 3, pp. 752–765, 2014.

[19] J. Morak, V. Schwetz, D. Hayn, F. Fruhwald, and G. Schreier, “Electronic data capture platform for clinical research based on mobile phones and near field communication technology.,” *Conference proceedings: Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society*. vol. 2008, pp. 5334–5337, Jan. 2008.

[20] J. Morak, H. Kumpusch, D. Hayn, R. Modre-Osprian, and G. Schreier, “Design and evaluation of a telemonitoring concept based on NFC-enabled mobile phones and sensor devices.,” *IEEE Trans. Inf. Technol. Biomed.: a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 16, no. 1, pp. 17–23, Jan. 2012.

[21] J. Morak, D. Hayn, P. Kastner, M. Drobics, and G. Schreier, “Near Field Communication Technology as the Key for Data Acquisition in Clinical Research,” *2009 First International Workshop on Near Field Communication*, pp. 15–19, Feb. 2009.

[22] G. Arfaoui, S. Gharout, J. Traore, "Trusted Execution Environments," in *Mobile Cloud Computing, Services, and Engineering (Mobile Cloud): 2014 International Conference,* pp. 259-266, 2014.

[23] KM. Alam, J. Kamruzzaman, “Dynamic adjustment of sensing range for event coverage in wireless sensor networks,” in *Journal of Network and Computer Applications:* Elsevier, vol.46, pp 139-153, November 2014.

[24] R. Di Pietro and J. Domingo-Ferrer, "Security in Wireless Ad Hoc Networks", *Mobile Ad Hoc Networking,* pp. 106-153, 2013.

[25] "RFC 2104 - HMAC: Keyed-Hashing for Message Authentication", Tools.ietf.org, 2016. [Online]. Available: https://tools.ietf.org/html/rfc2104. [Accessed: 21 May 2016].

[26] "Data Encryption Standard", www.tutorialspoint.com, 2016. [Online]. Available: https://www.tutorialspoint.com/cryptography/data_encryption_standard.htm. [Accessed: 01- Nov- 2016].

[27] "POPI Compliance", *POPI Compliance*, 2016. [Online]. Available: https://www.popi-compliance.co.za/. [Accessed: 01- Nov- 2016].

[28] "Ear Thermometer IR20 - ForaCare Ear Thermometer, and Diabetes Management Supplies", *Foracare.com*, 2016. [Online]. Available: http://www.foracare.com/Thermometer-IR20.html. [Accessed: 01- Nov- 2016].

[29] "Blood Glucose Monitoring System - G31 - ForaCare", *Foracare.com*, 2016. [Online]. Available: http://www.foracare.com/glucometer-G31.html. [Accessed: 01- Nov- 2016].

[30] "Blood Pressure Monitoring System - P20 - ForaCare", *Foracare.com*, 2016. [Online]. Available: http://www.foracare.com/Blood-Pressure-P20.html. [Accessed: 01- Nov- 2016].

[31] A.Mxoli, N. Mostert-Phipps, "Personal Health Records: Design Considerations for the South African context", Nelson Mandela Metropolitan University, Port Elizabeth, 2014.

[32] T. Mudaly, D. Moodley, A. Pillay and C. Seebregts, "Architectural frameworks for developing national health information systems in low and middle income countries", *Proceedings of the First International Conference on Enterprise Systems: ES 2013*, 2013.

[33] D. Moodley, C. Seebregts, A. Pillay and T. Meyer, "An Ontology for Regulating eHealth Interoperability in Developing African Countries", *Foundations of Health Information Engineering and Systems*, pp. 107-124, 2014.

[34] South African Department of Health, "mHealth Strategy", South Africa, 2015.

[35] C. Seebregts, P. Barron and G. Tanna, "MomConnect: an exemplar implementation of the Health Normative Standards Framework in South Africa", *SA Publications*, 2016.