A BIOMETRICS BASED SECURE COMMUNICATION SCHEME FOR BLUETOOTH ENVIRONMENT

PUNEET SONI

A BIOMETRICS BASED SECURE COMMUNICATION SCHEME FOR BLUETOOTH ENVIRONMENT

by

Puneet Soni

A Thesis Submitted to the Faculty of
The College of Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Science

Florida Atlantic University
Boca Raton, Florida
May, 2007

A BIOMETRICS BASED SECURE COMMUNICATION SCHEME FOR BLUETOOTH ENVIRONMENT

By
Puneet Soni

This thesis (or dissertation) was prepared under the direction of the candidate's thesis advisors, Dr. Abhijit Pandya, Department of Computer Science and Engineering, and Dr. Hanqi Zhuang, Department of Electrical Engineering, and has been approved by the members of his supervisory committee. It was submitted to the faculty of The College of Engineering and Computer Science and was accepted in partial fulfillment of the requirements for the degree of Master of Science.

SUPERVISORY COMMITTEE:

Abhijit Pandya

Thesis Advisor, Dr. A. S. Pandya

Thesis Co-advisor, Dr. Hanqi Zhuang

Member, Dr. Imad Mahgoub

Chairman, Department of Computer Science and Engineering

Dean, College of Engineering and Computer Science

Dean, Graduate Studies and Programs

5.4.07
Date

ACKNOWLEDGEMENTS

I would like to thank the members of my committee, especially Dr. Abhijit Pandya, for his expertise, criticism, and patient guidance throughout this research. I am grateful to Dr. Hanqi Zhuang for guiding me in the direction of perfection and better understanding of biometrics related topics. Finally, I would like to thank my brother Saurabh Soni, sister-in-law Vibha Soni, my good friend Sridhar Tadicherla for their support during all this time. Equally important, I am indebted to the aforementioned people for their friendship.

This research was partially supported through a research grant by Motorola, Plantation, FL to Dr. Pandya entitled, "Motorola Test Vector project". It was also partially supported by a federal earmark grant entitled, "Secure Wireless Telecommunications".

ABSTRACT

Author:	Puneet Soni
Title:	A Biometrics Based Secure Communication Scheme for Bluetooth Environment
Institution:	Florida Atlantic University
Thesis Advisor:	Dr. Abhijit S. Pandya
Degree:	Master of Science
Year:	2007

A novel personnel authentication and verification system for devices communicating through Bluetooth protocol has been proposed in this thesis. Unlike existing verification systems which provide password or a PIN as a key, the system uses biometrics features as a key. In the implementation of the scheme, ridges and bifurcation based parameters are derived to generate a 128 bit Bluetooth pairing PIN. In this thesis a unique translational and rotational invariant feature set has been developed. These extracted feature data, unlike traditional systems which include the extracted data into payload, is used for device connection by generating the 128 bit PIN. The system performance is analyzed using the pairing PIN for inter-sample and intra-sample recognition. To validate the stability of the system the performance is analyzed with external samples which are not a part of the internal database.

Table of Contents

List of Tables

List of Figures

Chapter1

Introduction

1.1 Objective and Motivation

The offices of yesterday were a mesh of wired cables which often resulted in confusion with the exponential growth in peripheral devices. With so many devices like fax machines, printers, desktops, laptops, landline phones, the hazard and the clutter associated with the cables could not be overlooked. Bluetooth as a wireless network protocol provides a low-cost solution, for a wireless and cable-free work area. Most of the electronic devices which are widely and commonly at workplaces, home and commercial establishments are Bluetooth enabled. The Bluetooth devices when connecting to each other follow the Bluetooth protocol where two devices share a common pairing key. The pairing keys chosen are based on choice of the user and are easy to be hacked and use by impostors, which can lead to serious security issues in places where high level of security is required. Biometric features provide the capability of a secure authentication because of the uniqueness associated with them. Among the extractable and collectible features, fingerprints have acquired widespread acceptance because of the ease associated with their collectibility and uniqueness. Other biometrics features are also widely used, but have certain disadvantages associated with them.

A system which can combine the wireless connectivity of the Bluetooth with the user-specific advantage of biometrics can provide a solution for scenarios, where high level security is required. Suppose, a user enters a high security wireless area and he/she wants to access the database of the system. If he/she happens to be a legitimate user, he/she can pair with the database master device and become a part of the whole set-up. As opposed to the other valid and established systems, the system will use the biometric data to generate the pairing key between the two devices. Thus the devices connected are not only permitting a legal user, also the slave device can be synchronized as per the user logging in, into the system.

1.2 Related Work

In the present scenario, where the use of the Internet has increased by manifolds, breach of the personal information has become a major threat to the Internet security. There is an urgent need of Internet security at all stages where customer authentication is needed. Biometric recognition, as a means of personal authentication, is an emerging area focused on increasing security and convenience of use in applications where users need to be securely identified [Ortega 2004]. Biometrics has been applied with many other systems, to increase the level of security. One of the several fields of biometrics based identification is application of biometrics in a wireless network.

In [Tikkanen 2003] a review of all the available biometrics features is provided along with their feasibility in wireless systems is apprised. A study of the development of the wireless body area sensor network (BASN) has been discussed by Poon and Zhang. [Poon 2006]. In [Jain 2003] the biometric data is used for final authentication in a Bluetooth environment. In [Clancy 2003] there is a description on how a private key stored on a smartcard is used authentication in a networked environment. In [David 2003] another biometrics and smartcard based authentication approach is described. A study on the use of biosensors implanted in human body over wireless networks was discussed in [Cherukuri 2003]. Although [Dellutri 2005] provides an insight for handset pairing, there has been little work in terms of utilizing the biometric data for pairing of the Bluetooth devices.

Most of the studies, involve extraction of the data from the biometrics feature, encrypting it into the packet length data payload format (i.e. 2746 bit field in Fig. 1.1) and transmission over the required layer. Hence there is no stringent size requirement imposed on the biometric used for encoding. Fig 1.1 shows the packet format in Bluetooth protocol.

Access Code	Packet Header	Payload
72 bits	54 bits	0-2745 bits

Fig 1.1 Bluetooth Packet format

The traditional methods involving the secure authentication and verification utilize passwords or PINs which are easy to be forgotten. With Biometrics a personnel feature of an individual is stored for authentication/verification from the live scan. This curtails the need of remembering a long password by memory or to write it down, where it could be compromised. People have a heavy reliance on passwords, either deliberately or by error. The passwords can be disclosed to illegitimate users. Also a possible hacker might be watching a user at the time of authentication, to take advantage at a later stage. With Biometrics there is no chance of this happening, as it requires the physical presence of the authorized user.

Biometrics based passwords also limit the potential damage caused by the user habit of using the same passwords for a large number of applications and the subsequent access of this password by a wrong person. Also, a user is relieved of the headache of changing the passwords after every definite period.

Our suggested approach provides a unique approach where the Biometrics data is not appended into the Bluetooth payload. Instead, it is utilized for pairing the Bluetooth devices and hence have the advantages over other systems where the data field is appended into the payload (as shown in Fig 1.1).This pairing key can be directly applied into the Bluetooth handsets for a user device synchronization.

1.3 Problem Statement

The goal here is to implement a system which will help enhance the security in the wireless connection between devices which follow the Bluetooth protocol. Traditional biometrically implemented methods involve remote authentication and verification, where the user can be recognized after correct /incorrect match from the database. In the implemented system, the user information is entered for device pairing of two Bluetooth devices. The verification of the user as authentic user allows him/her to use to pair up his/her Bluetooth device with the master device. The utilization of the biometrically derived key for device pairing eliminates the use of PIN or the password, which are easy to be forgotten and misused by other hackers.

Since other approaches involve extraction of the data from the biometrics feature and encrypting it into the packet length data payload format, there is no size limitation imposed on the biometric used for encoding. On the other hand in our approach the biometrics data is not appended into the Bluetooth payload. Instead, it is utilized for pairing the Bluetooth devices which results into necessary step to reduce the biometric data to 128 bits.

1.4 Scope and contribution

A novel personnel authentication and verification system for devices communicating through Bluetooth protocol has been proposed in this thesis. For pairing between two devices a Bluetooth passkey or a PIN is entered into both the devices. PIN is used in the creation of a 128 bit initialization key with the help of E22 algorithm(as shown in Fig 2.5).

Our suggested approach provides a unique approach where the Biometrics data is not appended into the Bluetooth payload. Instead, it is utilized for pairing the Bluetooth devices and hence have the advantages over other systems where the data field is appended into the payload (as shown in Fig 1.1).This pairing key can be directly applied into the Bluetooth handsets for a user device synchronization.

During the study, various possibilities of converting the unique biometric data into a feature vector were considered. Based on various considerations a scheme was devised to compress the pixel data from a 260x300 (i.e. 78,000 pixels) fingerprint image to 128 bits. This meant 312kbytes (78,000 pixels x 4 bits/pixel) were compressed to 128 bits using rotation and translation invariant feature set which we propose in this thesis.

For performance analysis, extractable fingerprint features were derived from a group of different users. These features were then compressed into 128 bits, which is the length of the Bluetooth device pairing key. The device pairing keys generated were analyzed for inter-recognition between the templates of the sample database.

Additionally the stability of the system was analyzed with respect to 100 external samples in order to determine the performance when a hostile intrusion is attempted.

1.5 Organization of thesis

This thesis is organized into five chapters:

In Chapter 1, the objective and motivation for the thesis are stated. The general background developments and schemes which implement Biometrics based secure authentication schemes in a Bluetooth or any other wireless environment is reviewed. This section also includes the scope and contribution of the thesis, along with this description of the thesis outline.

In Chapter 2, key Biometrics and Bluetooth features relevant to the overall scheme for the secure authentication are reviewed. The section is divided into two halves; the first half describes the various Biometrics features available and the relevance and commercial utilization of the features. The second half discusses the Bluetooth system along with a short description of Bluetooth device connection protocol, which is pertinent to our overall scheme.

Chapter 3 describes the implemented scheme. The section begins with an overview of the study featuring the commercially available Bluetooth system which was a part of the study. The next sub-section describes the implemented scheme.

Chapter 4 describes the results obtained. Results of iterations are presented when executed under a range of specified system thresholds. The results are analyzed as to the effect of these thresholds.

Conclusions that can be drawn from the results are noted in Chapter 5. Possible future studies suggested by these results are also discussed.

Chapter 2

Study of Biometrics and Bluetooth

In this chapter we will review Bluetooth and Biometrics in order to provide the reader with basic concepts that will make the understanding of the system implementation easier.

2.1 Biometrics

Biometrics is the science of identification of an individual based on the measurements of his physical characteristics. ('Bio'= life + 'metrics' = measurement). Biometrics authentication or, simply biometrics refers to establishing identity based on the physical and behavioral characteristics (also known as traits or identifiers) of an individual such as face, fingerprint, hand geometry, iris, keystroke, signature, voice, etc [Jain 2004]. The behavioral features include speaker recognition, keystrokes and signature dynamics. The physical characteristics are iris scan, fingerprint recognition, hand geometry, retinal scan, DNA etc.

2.1.1 Use of Biometrics

A large number of systems require reliable personal recognition to either confirm or determine the identity of the individual who require service. Only legitimate users should access the services rendered by these systems [Jain 2004]. For example, usage of cell phones, ATM, access to secure areas should be restricted to only authorized people.

Biometrics systems are pattern recognition systems which can be utilized in two ways. In a one-to-one comparison between an individual and a stored biometrics, and in a one-to-many comparison between an individual and multiple biometrics on a database.

Biometrics has certain salient features which makes it useful for authentication. Some of these salient features are:

a) Universality:- Every person has characteristics.

b) Distinctiveness:- No two persons have the same characteristics.

c) Permanence:- The characteristics do not change over a period of time.

d) Collectibility:- The characteristics can be measured quantitatively.

A generic biometrics system is described below:

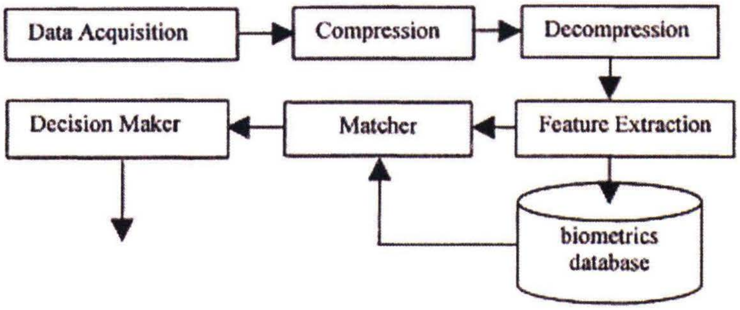


Fig 2.1: Generic Biometrics system

In this system, data is acquired using a device, like scanner, voice recorder, etc. The acquired data is used to extract quantifiable features. Some of the features like minutiae of fingerprints, unique vessel pattern of the iris, length of the fingers in hand geometry, etc are used for extractable features. The extracted data is compared with the database and depending on the result of a match or a mismatch, authentication or identification is performed.

2.1.2 Types of Biometrics

As was discussed earlier, there are several biometrics features which can be utilized for biometrics systems. In this section the biometrics features will be discussed in details and a brief commercial comparison of the commercially available features will be provided. [Anthony 2004]

Fingerprint:- Fingerprints are the most widely used biometrics systems. Fingerprints are widely used by FBI for secure authentication. A detailed study of fingerprints will be discussed later in this chapter.

Voice Recognition:- A popular methodology for implementing voice recognition would involve recording a user's voice over a given system and then comparing them with a database. Voice recognition systems have 1 to 2% False Acceptance Rate FAR [Woodword 2003] and are not robust on their own, but can be combined with other systems to provide lower FAR.

Iris Recognition:- Iris Recognition based models are unique, invariant and have higher confidence than fingerprints. However iris recognition was not widely used so far because of the patent issues. The patents on iris recognition technology have expired recently and they are supposed to become more popular.

Face:- Face recognition are often used for verification, as required in picture IDs[Shen 1999], as an easy means of identification for the eyes of humans. However, the human faces are known for ambiguity as two different persons might have same facial features. Also two twins may have same features.

Hand Geometry:- Hand geometry means the measurement of human hand geometry and is used by measuring the size of human fingers, knuckles etc. It can be combined with other biometric projects. It is not exclusive like finger printing and retinal eye scan.

The following figures provide a survey of the various biometric features in use.

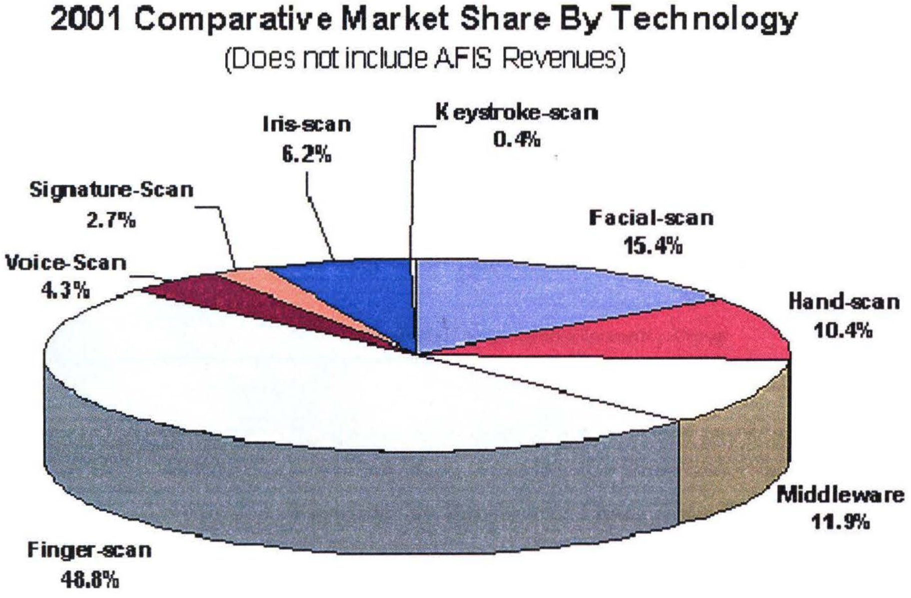


Fig 2.2 : Comparative commercial usage[Biometric Report]

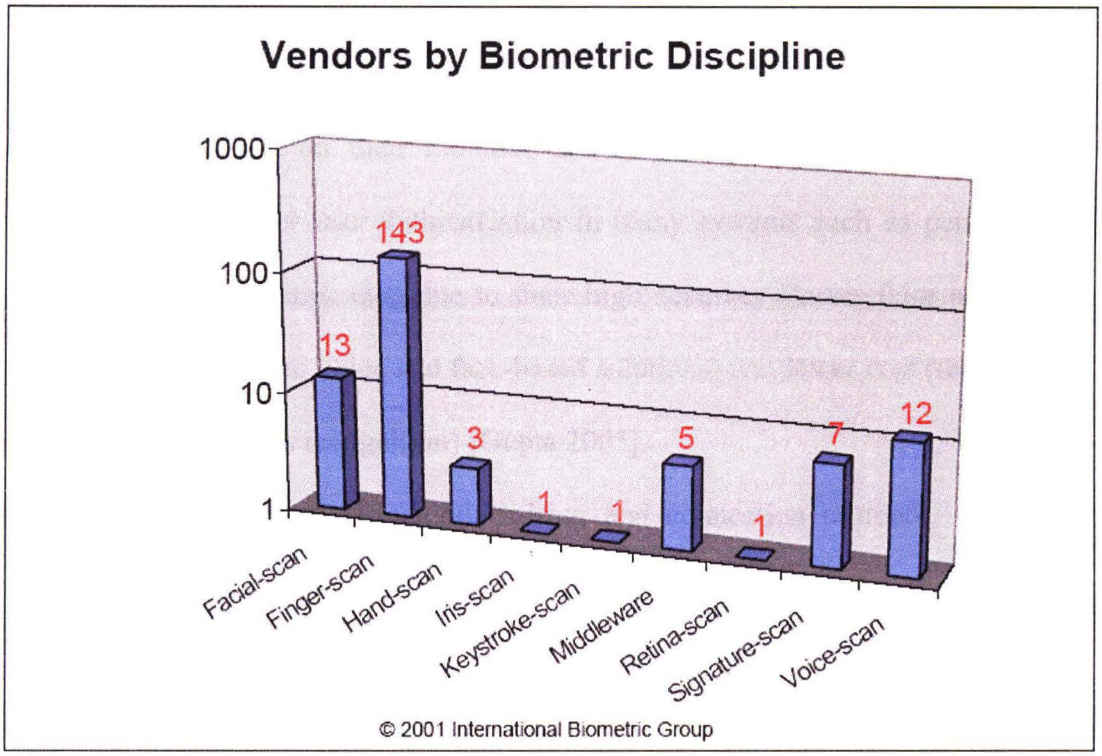


Fig 2.3: Vendors by Biometric Discipline [Biometric Report]

Note: A middleware vendor produces software that lies between the core biometric technology, which performs sample capture, feature extraction, and biometric matching, and a given application, such as Windows NT, Novell, or a PKI system. Biometric middleware is designed to offer a deployer a variety of biometric and non-biometric authentication options, such that the deployer is not tied to a single core technology.

2.1.3 Why fingerprints for our study?

Fingerprints based biometrics authentications have been used extensively for authentication and verification. FBI has been using fingerprint based authentication for a long time. One of the reasons for the widespread use of fingerprints is their uniqueness. This is primarily because of the ease of collection and highly unique structure of the fingerprint [Mennen 2005]. Fingerprints can be easily classified based on their minutiae and ridges. Also, fingerprint based systems can be used for user authentication in many systems such as personal laptops, enterprise systems, etc., due to their high accuracy (lower false accept and reject rates relative to voice and face-based solutions) and lower cost (relative to techniques such as iris recognition) [Gupta 2005].

The next section discusses Bluetooth pairing and connection protocol, which forms the basis of our devised scheme.

2.2 Bluetooth

Bluetooth is a cost effective solution for wireless connectivity. The initial application of the Bluetooth system involved has been regarded as a promising solution to an inexpensive wireless connection. Although initial application of Bluetooth technology has been focused mainly on replacing cables between hand-held devices, general wireless telecommunication such as public Internet access via a Bluetooth-equipped device is expected to be one of the most popular applications in the near future [Limt 2001].

In general, Bluetooth is the name of the short-range protocol operating at 2.4 GHz and is capable of functioning as a wireless connection. The range of this connection is 32 feet (10 meters). The data transfer takes place at the speed of 1 Mbps which is three to eight the speed of parallel and serial ports.

The Bluetooth SIG or the Bluetooth Special Interest Group is a privately held non-profit trade association, whose members are around 7000 leading companies worldwide which are pioneers in their fields like music, computing, telecommunications.

As per the SIG the maximum bandwidth permitted is 10Mb to a range of 10 meters operating at a range of 2.4 GHz - 2.483 GHz. Bluetooth can operate at three power classes:

a. Class 3 Devices which can operate at 100mw for a range of 100meters.

b. Class 2 Devices which can operate at 10mw for a range of 10meters.

c. Class 1 Devices which can operate at 100mw for a range 0.1-10 of meters.

There are three levels associated with Bluetooth:

a. Non-secure level, no security is implemented.

b. Secure level security, access is granted to individual services.

c. Link-level security, security is enforced at common.

In [Patricia 2004] the authors have described in details the Bluetooth protocol and how device connection is performed for a Bluetooth device. A Bluetooth Wireless Personal Area Network consists of Pico nets, with each Pico net consisting of around 8 Bluetooth devices. Each Pico net has 1 master device and all the other devices are the slave devices.

Fig 2.4 gives a description of how devices can be connected in a Bluetooth environment.

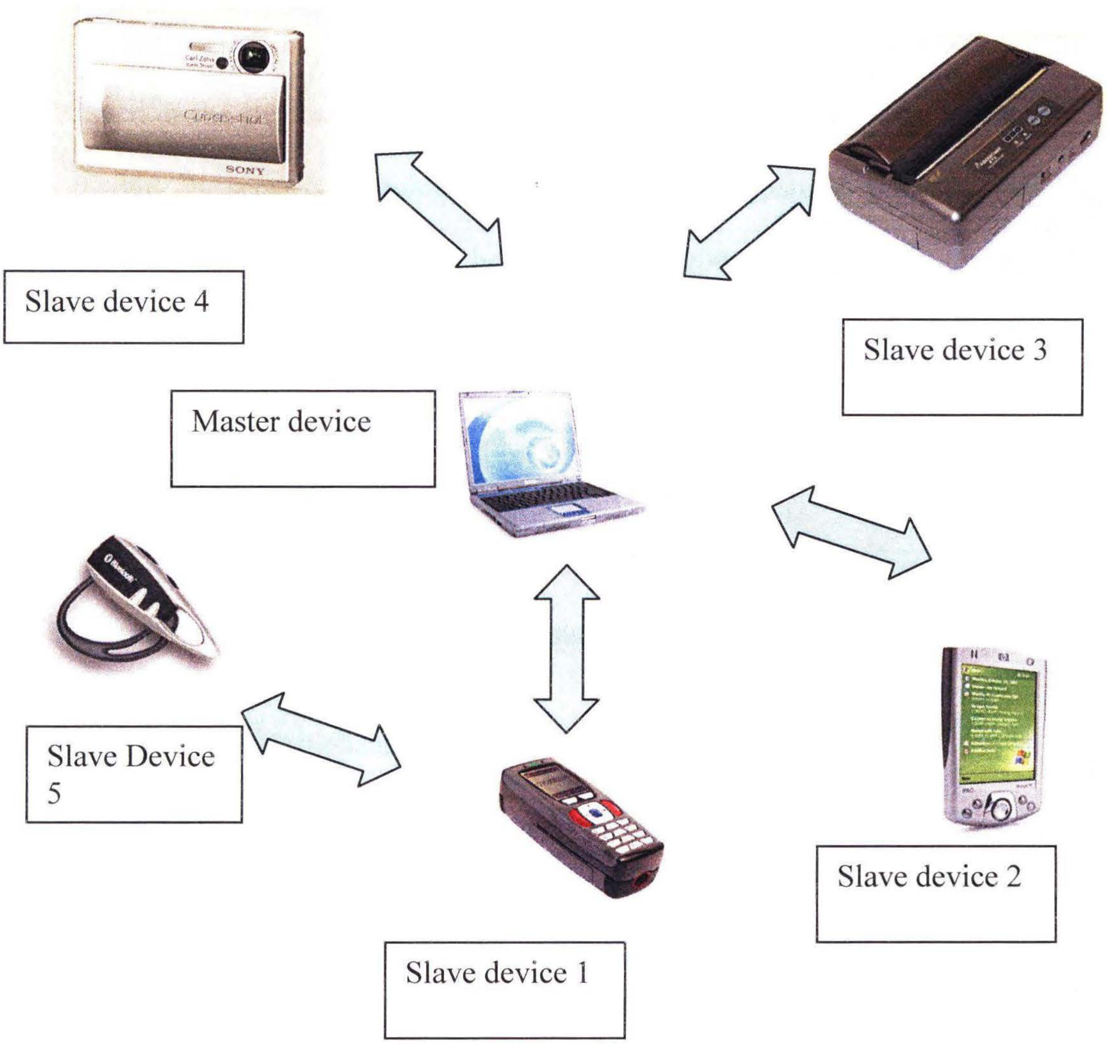


Fig 2.4: A Bluetooth Pico net.

As per Fig 2.4, in a wireless Bluetooth environment, the master device is connected to two or more slave devices. The master device searches for other peer devices for pairing.

Security protocols for Bluetooth Pico nets, such as key-management and authentication have been defined by the Bluetooth Specification [Bluetooth_Specification].

In this section the device pairing in Bluetooth is reviewed to provide a better understanding.In security terminology, establishing a channel between two Bluetooth devices is called pairing or bonding [Yaniv 2005].

2.2.1 The Bluetooth pairing and authentication process

The Bluetooth initialization procedure consists of 3 primary steps:

1. Creation of an initialization key (K_{init}).
2. Creation of a link key (K_{ab}).
3. Authentication.

For pairing between two devices a Bluetooth passkey or a PIN is entered into both the devices. For headset connection devices, the pairing key is fixed and has to be entered into the master device.

2.2.1.1 Creation of K_{init}

The creation of the K_{init} is done with the help of E22 algorithm. The algorithm produces a 128 bit output number, i.e., the K_{init}. The E22 algorithm has the following 3 inputs:

1. A *BD_ADDR*, Bluetooth device address.
2. The PIN code and its length.
3. A 128 bit random number *IN_RAND*.

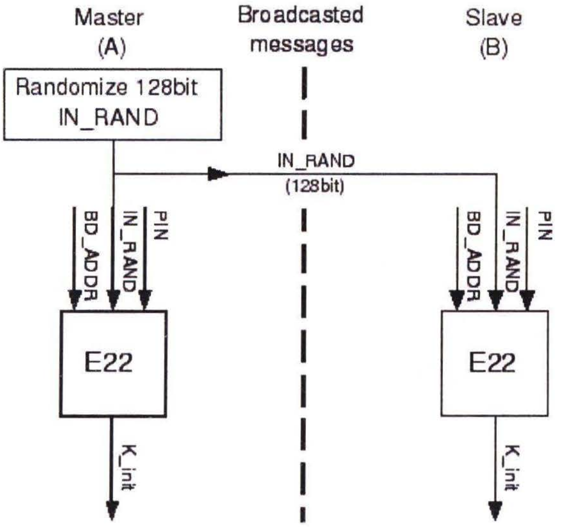


Fig 2.5: Creation of K_{init}

As shown in Fig 2.5, the three inputs are used for the generation of K_{init}. The K_{init} will be different for both the devices.

2.2.1.2 Creation of K_{ab}

The K_{ab} generated as explained in the previous section is used for the derivation of the K_{ab}. The device uses the initialization key to generate the K_{ab}. Each device sends the other device a random number, which is Xored with the K_{ab}. This input is used to in the E21 algorithm for the generation of K_{ab}, along with the BD_ADDR.Fig 2.5 gives an outline for the creation of K_{ab}.

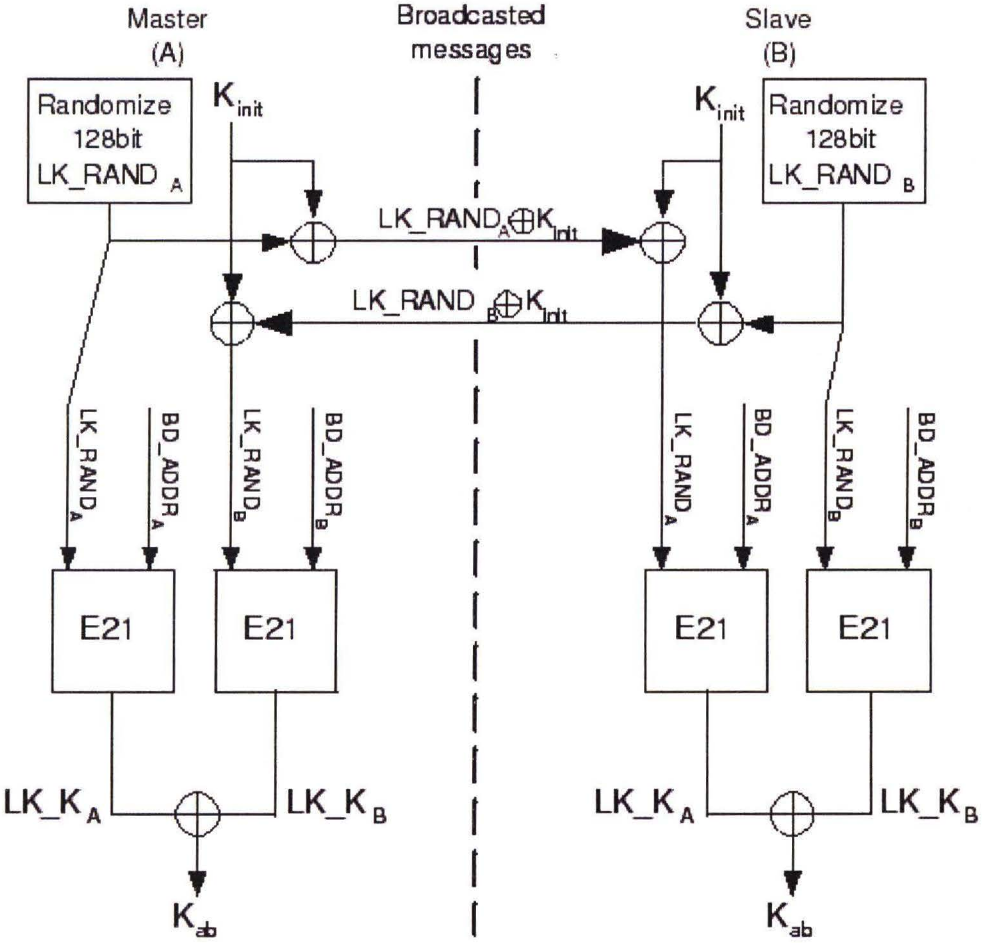


Fig 2.6: Creation of K_{ab}

Upon creation of the link key K_{ab}, mutual authentication is performed.

In [Yaniv 2005] the authors have described a way to crack the Bluetooth PIN. But there has been little research on utilizing other data, other than manually generated PINs for making Bluetooth device pairing more secure. The next chapter will provide an insight on the implemented approach.

Chapter 3

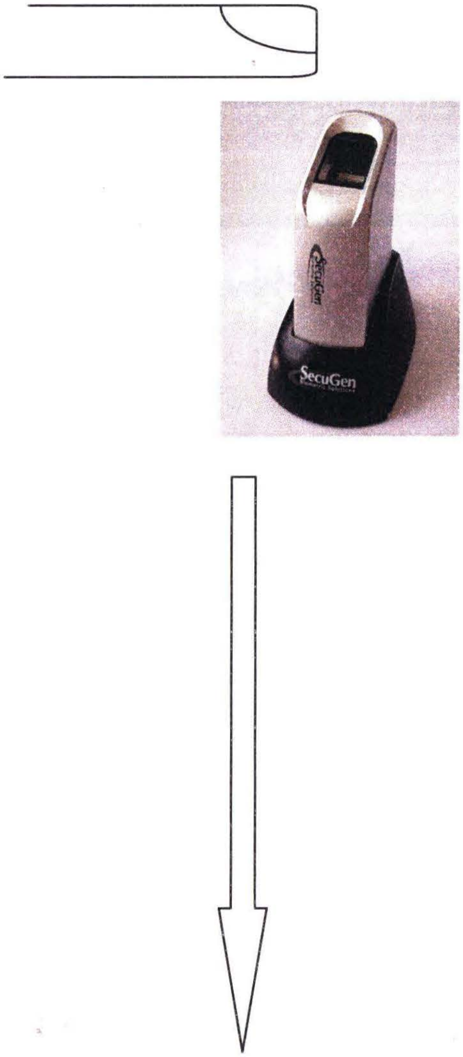
Proposed Biometrics Based Secure Communication System

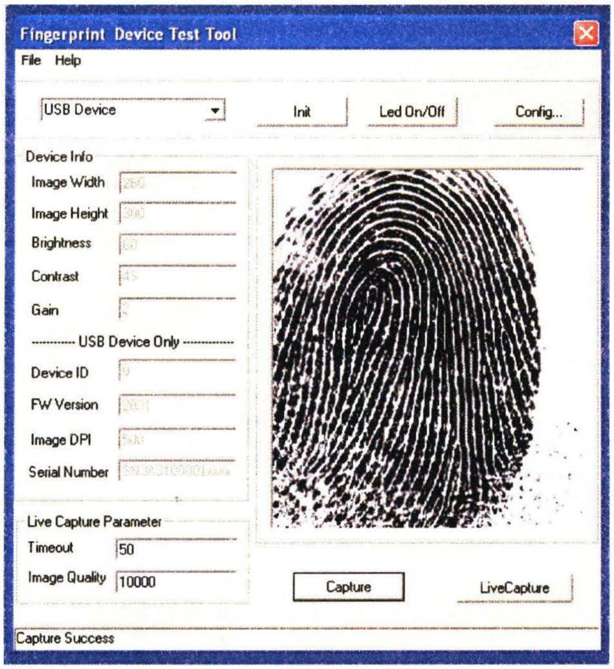
In this chapter, the various feasible approaches which were explored in order to design a biometric based communication system for Bluetooth authentication will be discussed. The most important part was to convert the data obtained from the fingerprint into 128 bits, which is the size of the encryption key in the Bluetooth payload (refer Section 3.3). This issue involves data compression with a loss of information. The scanned fingerprint image is 260X300=78,000 pixels with 4 bits/pixel representation, i.e., 16 gray levels, which is 312kb. Hence direct compression is very difficult and it is necessary to reduce raw data features to informative features like minutiae, ridges, bifurcations. In the first section, we will explain the system through which the fingerprint data was extracted. The next section will discuss the various feasible systems which were explored as a part of the study. The final section will discuss the proposed scheme for this encryption.

3.1 System Description

In this section the set-up for the whole system has been discussed.

The raw data from the fingerprint was obtained by taking multiple scans of the same finger using a Secugen® optical fingerprint scanner with a resolution of 260 x 300 pixels.





USB Device

Device Info

Image Width: 260

Image Height: 300

Brightness: 60

Contrast: 45

Gain: 2

---------- USB Device Only ------------

Device ID: 0

FW Version: 2001

Image DPI: 500

Serial Number: [illegible]

Live Capture Parameter

Timeout: 50

Image Quality: 10000

Capture Success

Feature extractor

Output image data

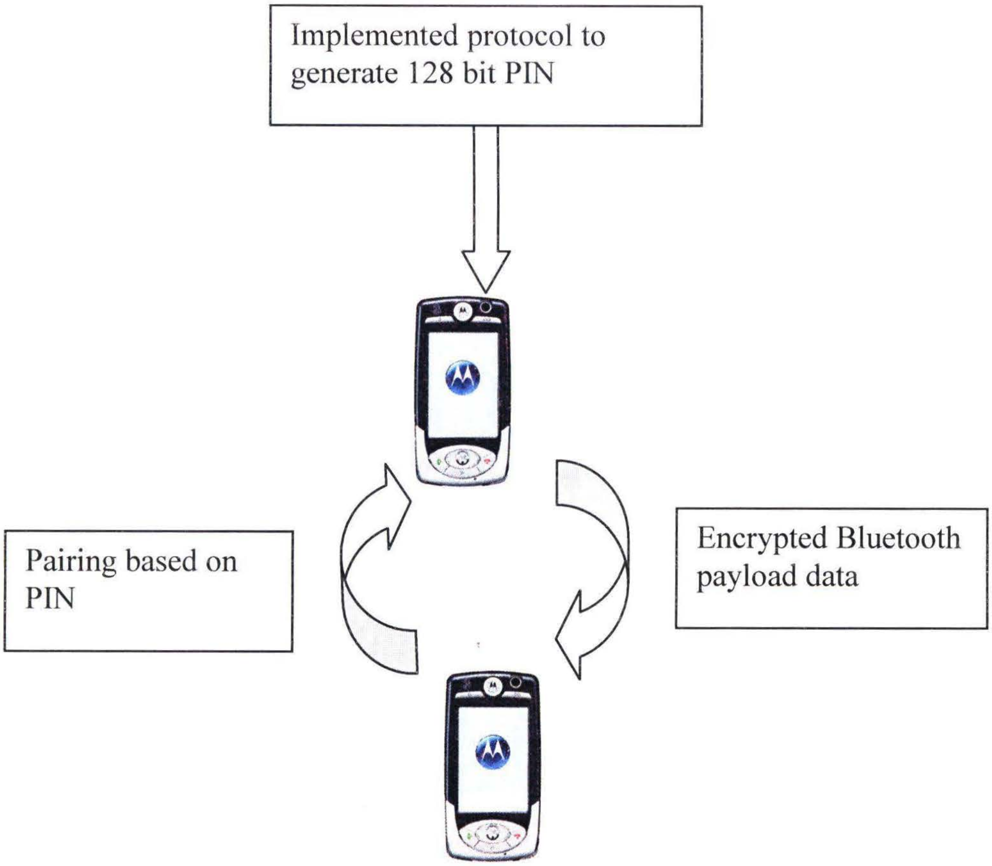
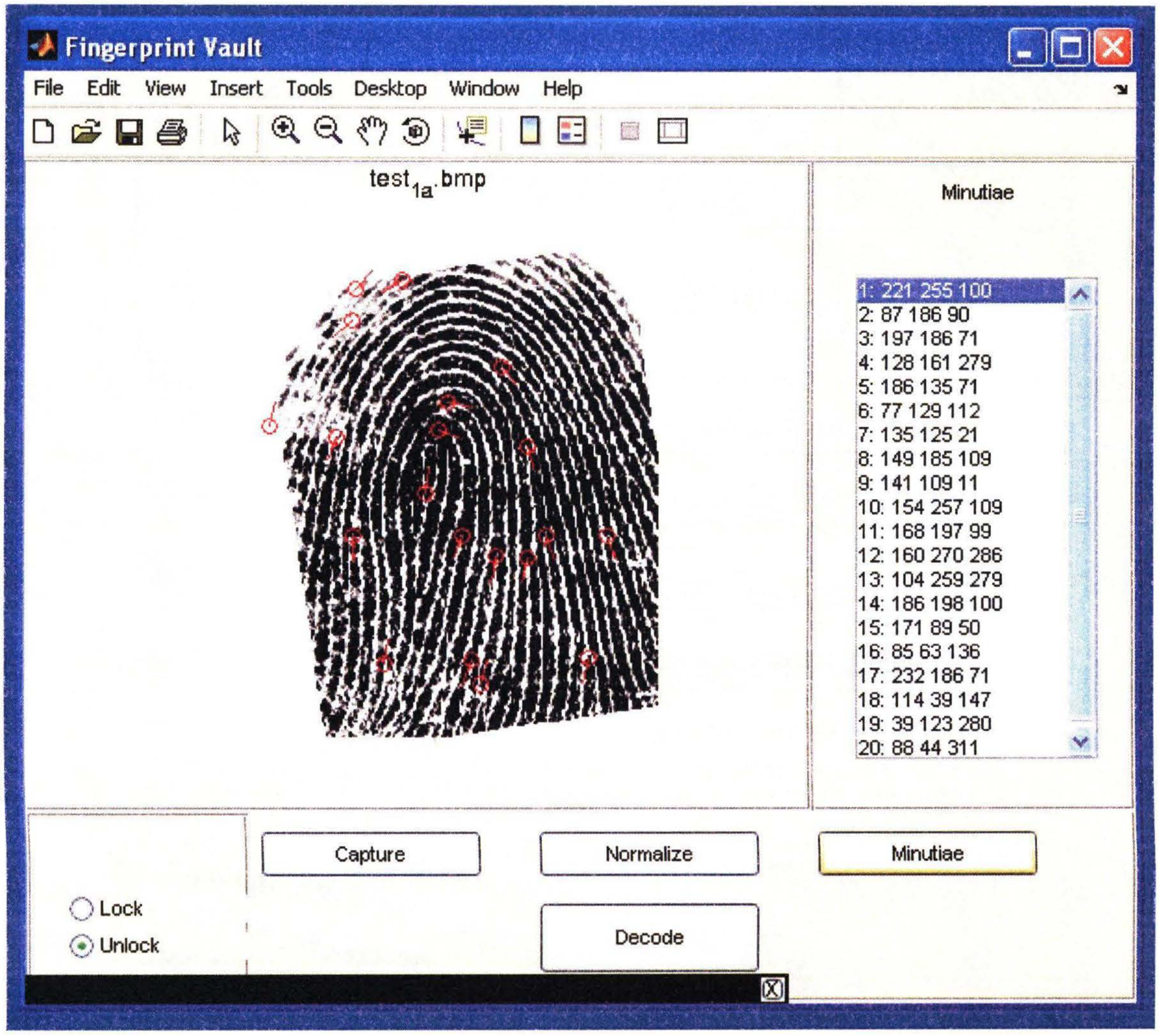

Implemented protocol to generate 128 bit PIN
Pairing based on PIN
Encrypted Bluetooth payload data

Fig 3.1: Implemented System set-up

Along with the output image containing the marked minutiae points, the feature extractor also generates a file which contains the number and the location of the fingerprints in the input sample.



```
1: 221 255 100
2: 87 186 90
3: 197 186 71
4: 128 161 279
5: 186 135 71
6: 77 129 112
7: 135 125 21
8: 149 185 109
9: 141 109 11
10: 154 257 109
11: 168 197 99
12: 160 270 286
13: 104 259 279
14: 186 198 100
15: 171 89 50
16: 85 63 136
17: 232 186 71
18: 114 39 147
19: 39 123 280
20: 88 44 311
```

Fig 3.2: Minutiae extraction.

In the already normalized fingerprint image shown in the left pane, identified minutiae points are shown in red. The right pane lists these points individually in the following format: minutia number, *x*-coordinate, *y*-coordinate, theta (angle).

The following table is a tabular representation of the one of the fingerprints of the database.

X co-ordinate	Y co-ordinate	Angle	Type of minutiae	Random number
161	125	56	B	60
31	266	210	B	60
28	278	52	B	60
45	280	228	R	60
216	242	286	R	60

Table 3.1: The output file format generated by the feature extractor

The data obtained in the minutiae reading from the fingerprint has 4 fields:

1) X co-ordinate- the first field.

2) Y-co-ordinate- the second field.

3) A, the angle the minutiae makes with respect to the origin.

4) The type of the minutiae i.e., ridge type (R) or bifurcation (B).

3.2 Feasibility system

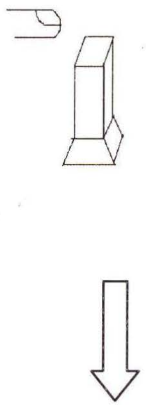
The following section will discuss the research which was applied to study the feasibility of existing commercial systems.

3.2.1 Belkin Bluetooth device

One of the early approaches which we adopted for achieving the goal was the use of Belkin Bluetooth devices. The Belkin devices consisted of two USB devices (Fig 3.3).

Fig 3.3: A commercial Belkin USB

Two Bluetooth adapters when connected over two different computer systems serve as a Bluetooth Pico net.



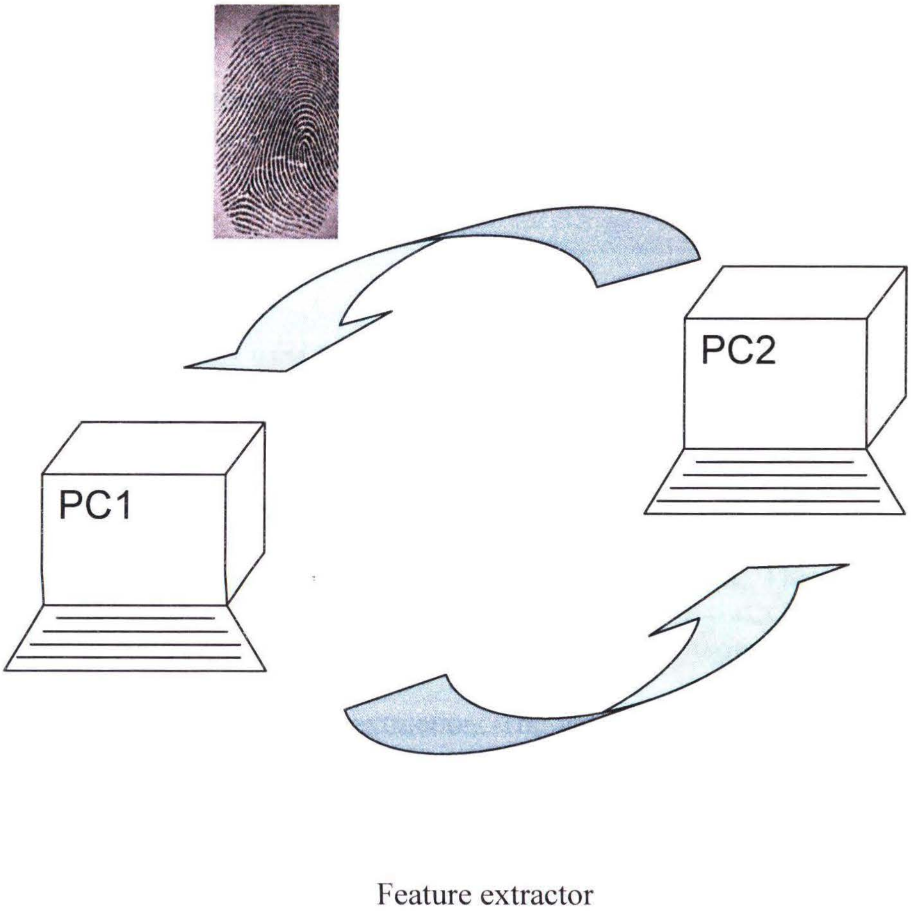


Fig 3.4: The Belkin device Set-Up

Referring to Fig 3.4, the fingerprint from the finger is obtained by the scanner. The information obtained from the fingerprint was stored in first database system PC1. PC2 had the stored information for the incoming fingerprints.

The system did not convert the fingerprint data into Biometric PIN, which needs to be provided for Bluetooth protocol communication system. Instead, it used the entire fingerprint data and plugged it into data field of the Bluetooth payload and then used its own algorithm for providing secure communication using that part of data field, and hence did not meet our stated objective (as was shown in Fig 1.1).

3.3 The Implemented approach

In this section, the scheme which was used for the final analysis of the studies has been reviewed .This new method is based on rotation and translation invariant features of the fingerprint. The number of ridges and bifurcations of the fingerprints are invariant to the rotation and translation of the fingerprint. The distance and the number of ridges and bifurcations is neither altered nor tempered when the fingerprint samples are utilized for extraction. The data extracted from these features are applied in study of biometric characteristics of False Acceptance Rate and False Rejection Rate.

The first step involved in recognizing the features which would be independent of translation and rotation of the fingerprint. The uniqueness of the fingerprint is dependent on the ridges and bifurcations. Total 6 parameters i.e., total number of minutiae, total number of ridges, total number of bifurcations, maximum distance between minutiae, maximum distance between ridges and maximum distance between bifurcations of every person, were analyzed in the implemented approach.

The samples chosen are the right hand thumb for every person. Total numbers of samples chosen are 10 per person. There are 11 people in the study. Therefore the total numbers of samples are 110.

First step was to calculate the "mean value" of all the 6 parameters per person. The logic behind the thinking is that since the samples were collected from the same fingers the "mean value" should not vary too much for intra-person recognition.

Table 3.2 shows a matrix which contains the information of all the 11 people with all the 6 parameters. The first column is the total number of minutiae, the second column is total number of ridges, the third column is the total number of bifurcations, the fourth column is the maximum distance between minutiae, the fifth column is the maximum distance between ridges, and the sixth column is maximum distance between bifurcations of every person.

Template ↓	No. of Minutiae	No. of ridges	No. of Bifurcations	Distance Between minutiae	Distance Between ridges	Distance Between bifurcations
Template1	75	45	30	340	327	331
Template2	25	10	15	330	286	326
Template3	46	21	25	339	333	322
Template4	27	14	13	334	320	313
Template5	48	29	19	347	339	319
Template6	35	17	18	327	303	316
Template7	35	18	17	342	341	326
Template8	46	11	35	335	300	329
Template9	21	3	18	311	246	307
Template10	56	30	26	353	352	316
Template11	38	12	26	347	336	337

Table 3.2: The table with the values of the templates with their respective values

The next step involves calculating the difference of every fingerprint's 6 parameters with their corresponding parameters from the above mentioned matrix. The following computation was used to arrive at the results:

$$\sum_{i=1}^{6} \frac{|X_i - T_{ij}|}{T_{ij}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{eq(1)}$$

Where, X=incoming fingerprint file,

T=Matrix corresponding to the fingerprint

i= parameter of the fingerprint

j= No of samples

As mentioned before, T_{ij} contains the information about the average of all the six parameters obtained for every single person. The logic behind the implementation is that since the information obtained is the average of the same fingerprint sample, there should not be much variation per person. Therefore the normalized distance between the same person's samples with its own template should be very small value. Since the fingerprints are a unique feature of a single person, these distances should also have the unique information about the person, i.e., they should be smallest for the same template.

The next step involved in calculating the threshold values for all the samples. False Acceptance Rate (FAR) and False Rejection Rate (FRR) are the functions of threshold values.

As was mentioned already the normalized distance for the same fingerprint samples with their own average, should be the least. The normalized distances were chosen as threshold values. For lower threshold values, only the lower distances should be accepted by the system and there will be a lot of false rejection. For higher threshold values, there will be a lot of false acceptance.

In the further implementation, 100 more fingerprints were extracted from different people. The 100 external fingerprints were registered in the system with the help of the 11 templates. The normalized distance of the external fingerprints was obtained with reference to the 11 templates. The next step was to decide, whether to accept the fingerprint or to reject it. Threshold value was the parameter chosen for this decision. The samples were analyzed for the randomly generated threshold values and then the total number of samples which were "accepted" and "rejected" on the basis of these threshold values. It may be pointed out here that these fingerprint values were "accepted" and "rejected" only hypothetically. Since these fingerprints were from external sources there can be no way they can be classified as from authentic users. The normalized calculated distances generate some values which happen to be very small. These small values may be very close to one of the templates and the choice of threshold value groups the fingerprint with one or the other template. The iterations were run for each fingerprint and their threshold values, were calculated.

The next step lied in getting the data from the fingerprints into 128 bit. As was mentioned already mentioned that the PIN utilized in generation of the passkey for Bluetooth devices can be anything from 0-128 bit. The complexity involved in generating 128 bit compression of the amount of fingerprint information has been discussed before. It will require a compression ratio of around 16-1, to simply compress the data of the fingerprint. The study could involve plain compression and then implementation, which would not, contributed to the study much.

A simple process was implemented for the study. A simple approach was generated for utilizing the data obtained from the study, into a string of size of 128 bits or lesser.

The 6 parameters were concatenated into a string with the first 2 bytes being occupied with the number of minutiae. The next 2 bytes with the number of bifurcations, the next 2 with the number of ridges. This data generated is similar to the one extracted in the previous studies.

```
        75          45          30          340          327          331
←---1--→

←---------2------→

←-------------3--------------→

←-------------------4-------------------→

←----------------------5--------------------------→

←--------------------------------6-----------------------------→
```

Fig 3.5: The data in the 128 bit field

1) No. of Minutiae

2) No. of ridges

3) No. of bifurcations

4) Max distance between minutiae

5) Max distance between ridges

6) Max distance between bifurcations

In the assumed set up, the slave Bluetooth device searches for other Bluetooth device in its vicinity of 10 meters. After finding the master device which, it needs to connect, the Bluetooth devices try to pair up with each other with PIN. The 6 parameters from the incoming fingerprint file are concatenated as explained in the Fig. This will result in a PIN of size less than or equal to 128 bits. This PIN is entered into the slave device. The Bluetooth protocol asks for the PIN on the Master device. From the master device, the 128 template string is entered into the device. If the PIN entered on the devices are found to be matched, then the devices are connected.

The objective of the study was to convert the data from the fingerprint file into 128 bit PIN. The data thus obtained will have less than 128 bits and still has enough data space for appending more data.

Chapter 4

Results

The overall devised scheme as discussed in Chapter 3 was implemented through program written entirely in C language.

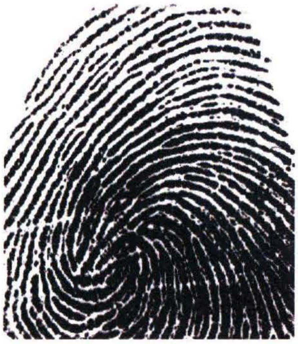
The following sections will describe the step-by-step analysis of the whole scheme.

The code can be found in the appendix.

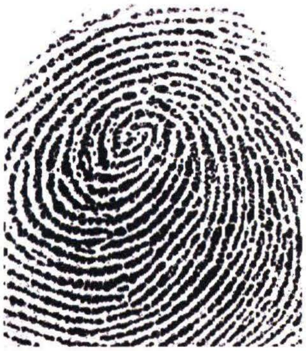
4.1 Database set-up

As discussed in Chapter 3, fingerprint database was collected from 11 different people.

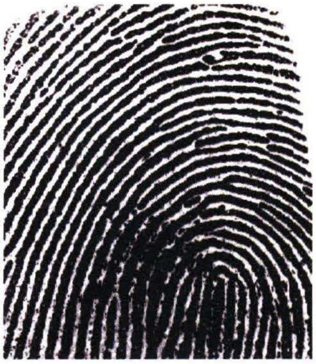
The 11 template constituting the database have been listed as per their order in the database. For all the 11 templates 10 prints of the same thumb were analyzed. The following Fig 4.1 shows the thumbprint which has the nearest value for its own subject.



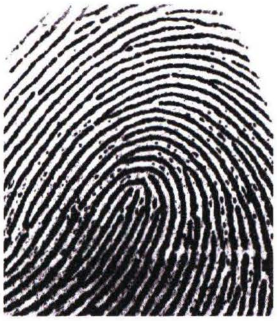
Subject pun



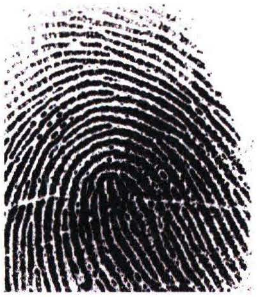
Subject drpandya



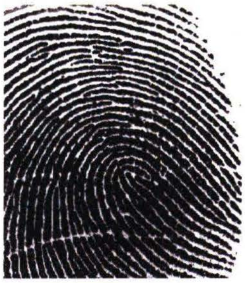
Subject yon



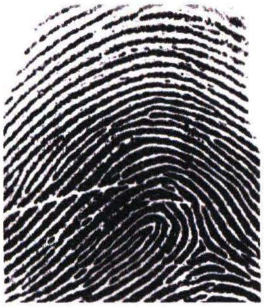
Subject s



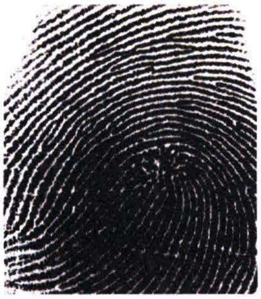
Subject v



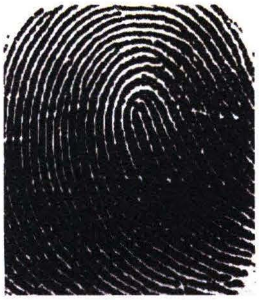
Subject su



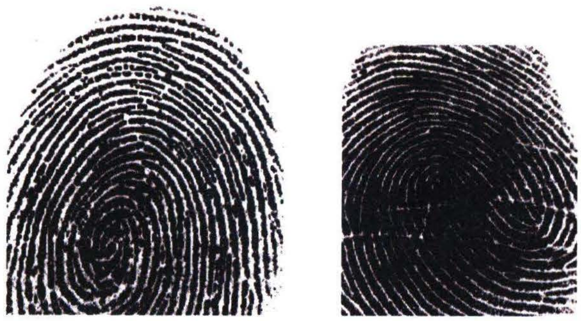
Subject sid



Subject pit



Subject nit

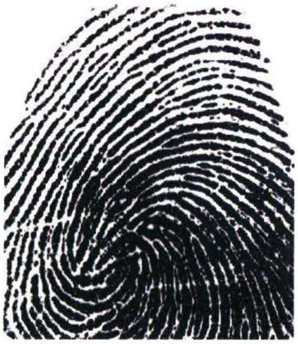


Subject henry Subject sri

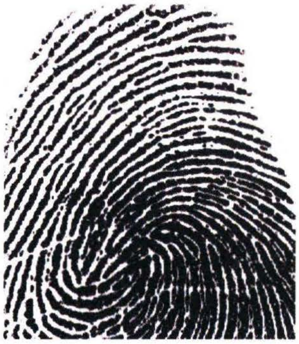
Fig 4.1: All the 10 templates of the study

4.2 Template 1

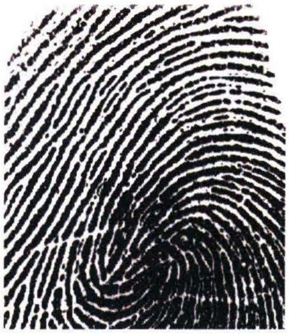
This section reviews the template 1 and all its 10 fingerprints. It was observed during the study that the fingerprints of subject 1 were somewhat unique, since it did not get misclassified as any other fingerprint template. Also none of its fingerprints were recognized as any other fingerprints. Thus we can state that it was an ideal example, since it was always correctly classified. The following Fig 4.2 contains all the 10 fingerprints of the template 1.



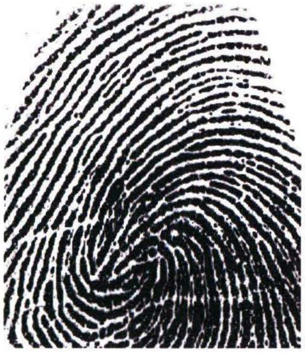
Fingerprint1



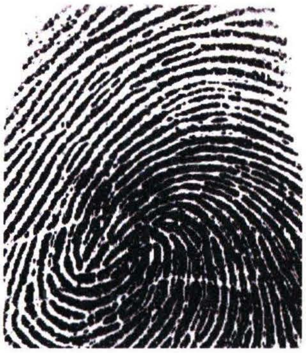
fingerprint2



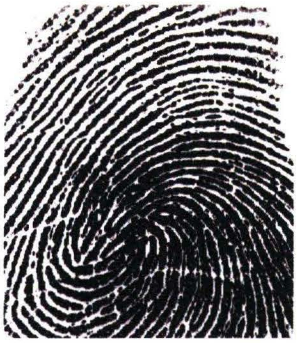
fingerprint3



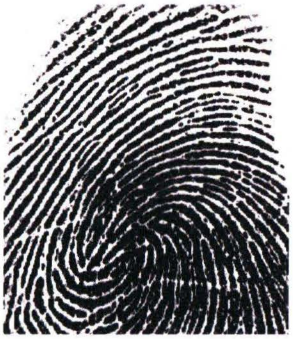
Fingerprint4



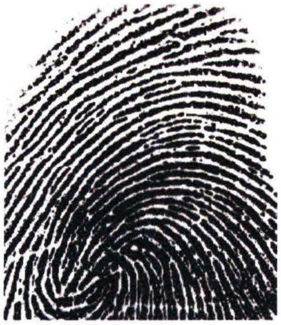
fingerprint5



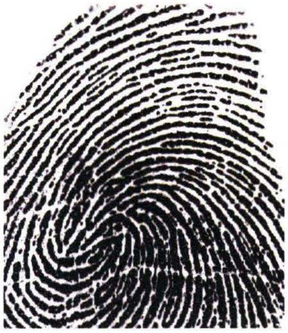
fingerprint6



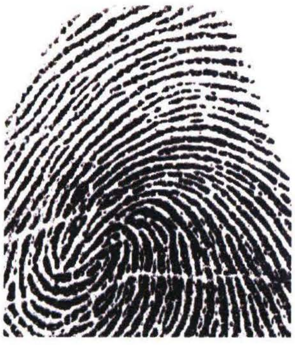
Fingerprint7



fingerprint8



fingerprint9



fingerprint10

Fig 4.2: All ten fingerprints of subject pun

4.3 Correct Acceptance and Wrong Acceptance

One of the major considerations of this study has been to establish, which samples were correctly accepted and which samples were wrongly rejected, to justify the system functionality. This section discusses the samples which were correctly accepted, and which one were wrongly accepted as other samples.

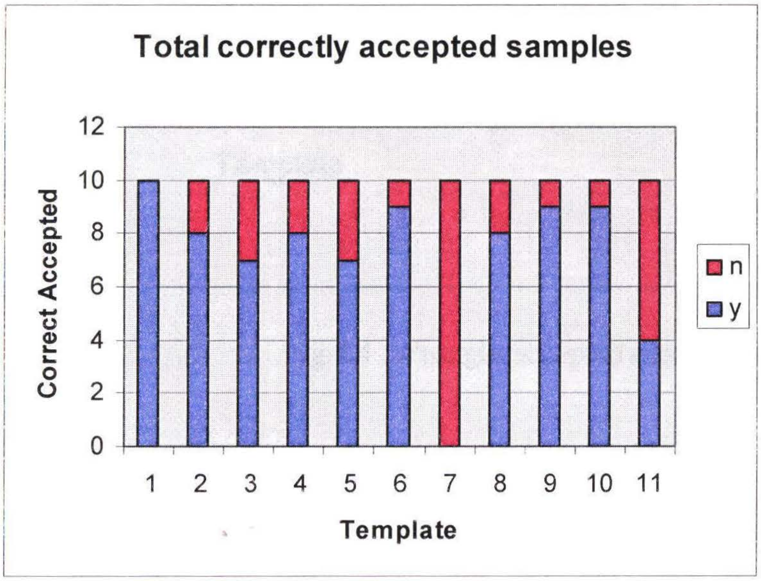


Fig 4.3: Total correctly accepted samples

Referring to Fig 4.3, the horizontal axis indicates the template number and the vertical axis indicates the total number of samples of the template in consideration. The bar-graph shows which fingerprints were correctly accepted or wrongly accepted as indicated by the blue bar. It also shows which fingerprints were wrongly accepted as another template (red bars). The anomaly associated with sample 7 will be discussed later in the chapter.

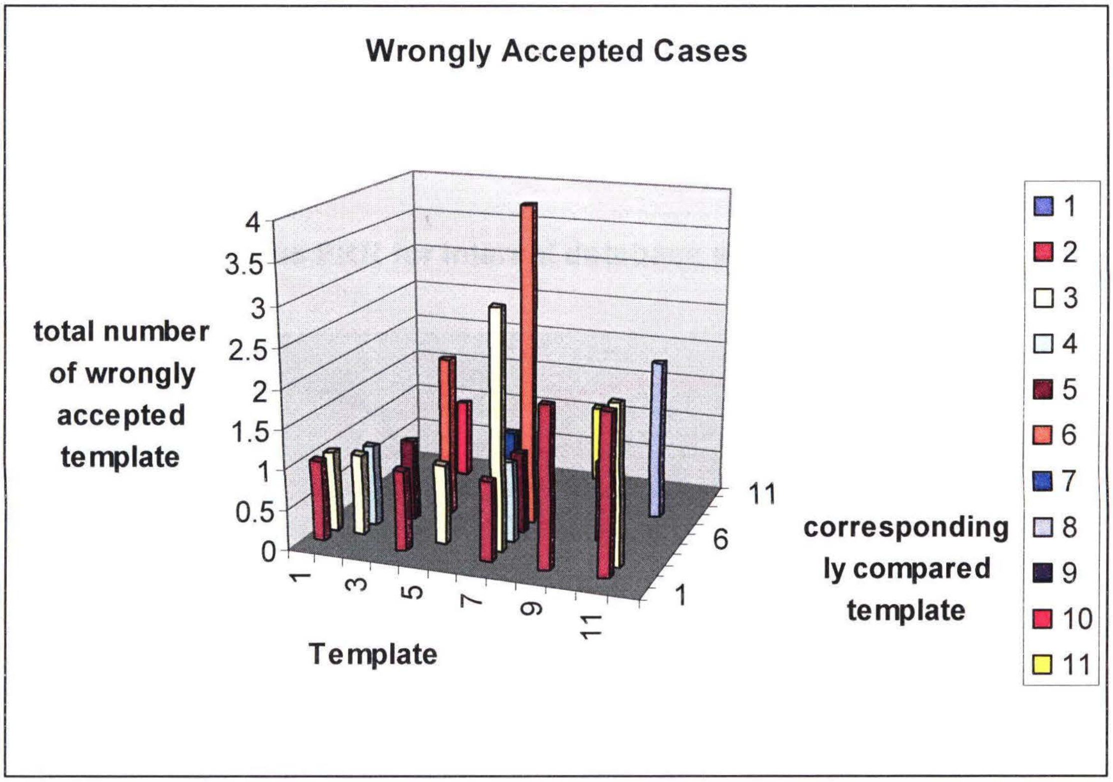


Fig 4.4 :Wrongly accepted cases

Fig 4.4 is further expansion of fig 4.3. The chart explains which template was misclassified as another template. The X –axis refers to the subject in consideration. The Y-axis refers to the template which it is being recognized as. The Z-axis refers to the number of misclassified samples. It can be seen from Fig

4.4 that template 6 has the largest number of misclassified fingerprints of template 7. Section 4.4 will discuss the reason for such a large acceptance of fingerprint 7 as subject 6.

4.4 The Variation of the threshold values for the Samples

This section analyses the effect of variation in the threshold value with respect to recognition value of samples.

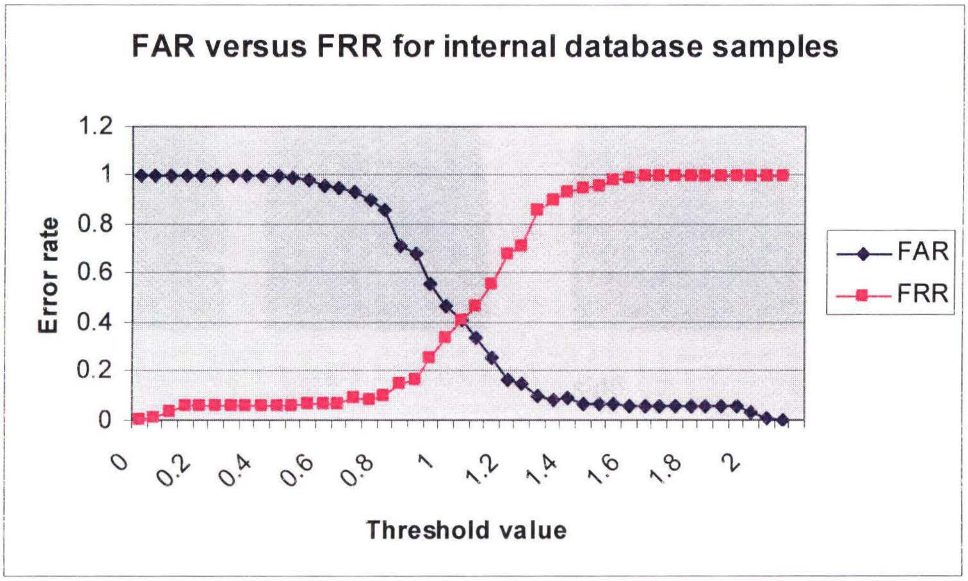


Fig 4.5: FAR versus FRR for the internal database samples

Fig 4.5 shows that initially FAR is constant till the value of threshold 0.6, after that it starts decreasing from 0.85.The Equal Error Rate (EER) for the internal samples occurs at a value of 1.05. The graph was plot without the erroneous fingerprints samples of 7 & 11.

4.5 Study of template 7

This section shows the template 7, all its fingerprints and we explain which one is being getting recognized as which one.

Sid1

sid2

sid3

Sid4

sid5

sid6

Sid7

sid8

sid9

sid10

Fig 4.6: shows the 10 fingerprints of subject 7

The close observation of any sample suggests presence of the arches and the delta together in the fingerprint template. This can be suggested as one of the reasons why this fingerprint is being recognized as some other template. Also since both template 7 and template 6 have the same number of minutiae, most of the template 7 fingerprints (4 fingerprints) are recognized as fingerprint 6.

4.6. A discussion on high recognition and authentication for template 1, 2 and 3

This section reviews the results for High value of True Acceptances of subject 1 and high rate of False acceptance for subjects 2 and 3.

Template 1 enlists 75 minutiae (Fig 3.7).All the other 10 templates have fewer than 75 minutiae. It can be suggested that for correct authentication a fingerprint with a large number of features has more probability for being correctly recognized.

For fingerprint template 9, there are only 3 ridges, and 18 bifurcations on an average.

the lower number of ridges or the absence of a large number of features, make it a unique finger print.

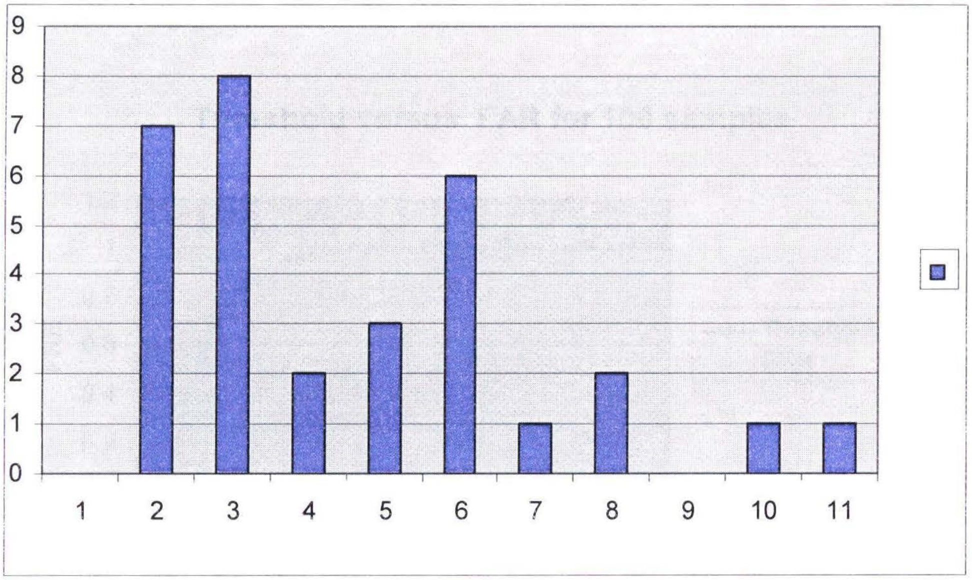


Fig 4.7: A study of the wrongly recognized subjects

As can be seen in this table Fig 4.7, template 2 & 3 are wrongly recognizing many of the samples of the other subjects as their own .The X-axis refers to the subjects in the study. The Y-axis refers to the total number of wrongly recognized fingerprints of the other fingerprints.

4.7 Study of 100 external samples

Another study included testing of 100 external samples with respect to the database. The system design allows co-operative users to be qualified as the authentic users. Since theoretically the 100 external samples were not from the database, the probability of their resembling the database samples should be very small. The iterations were started from a value of zero threshold, to a maximum value for which all the samples were accepted.

The following graphs show the FAR and the FRR for the 100 external samples.

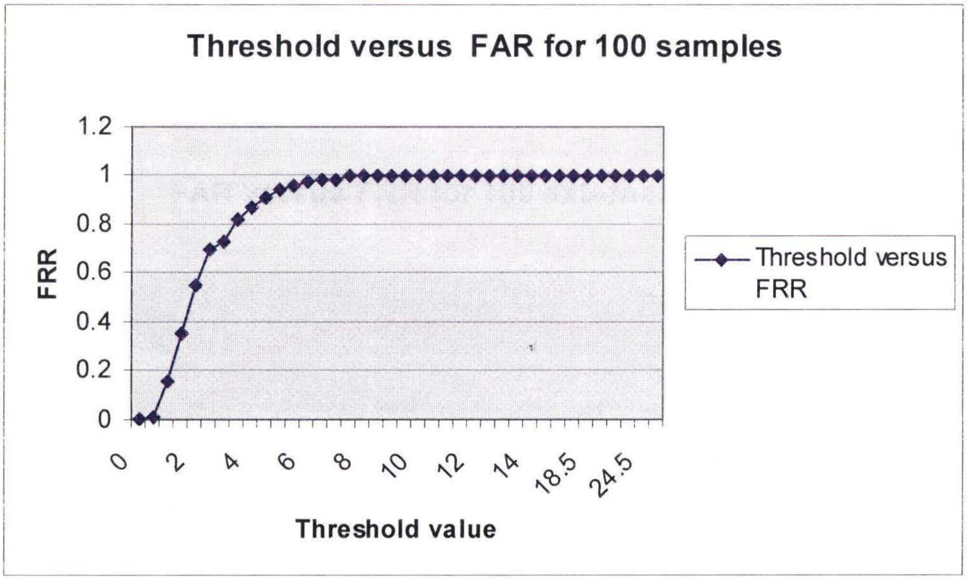


Fig 4.8: Threshold versus FAR for 100 samples

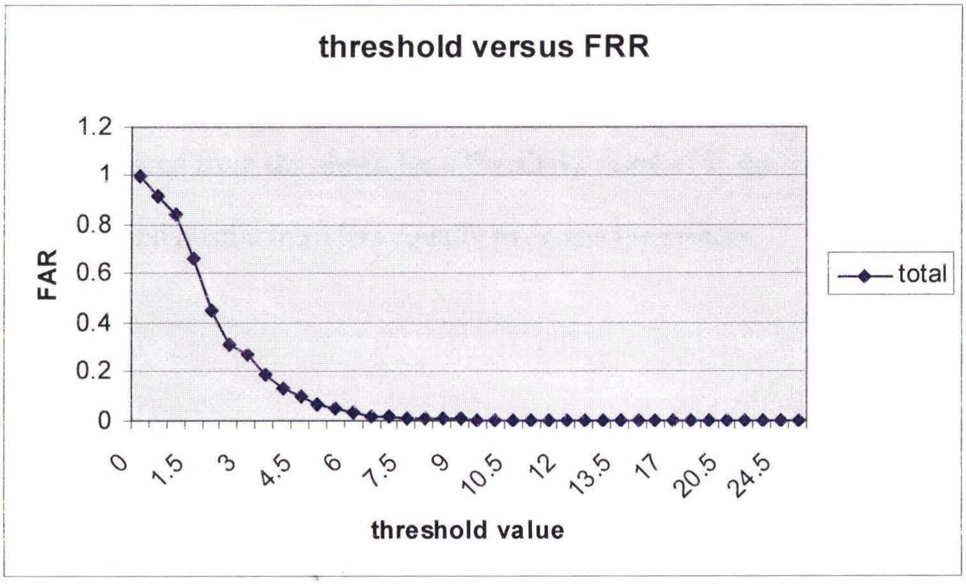


Fig 4.9: Threshold versus FRR for 100 samples

As can be seen in the figures 4.8 and 4.9 the FAR and the FRR are complementary to each other. With the increase in the threshold value the FAR decreases and the FRR increases. Fig 4.10 shows the Equal Error Rate for the 100 external samples.

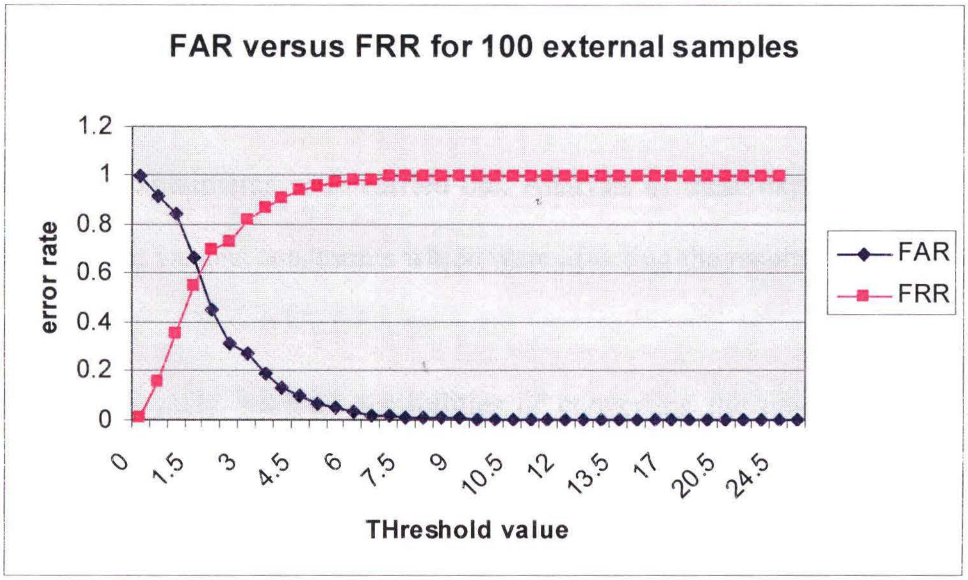


Fig 4.10 Equal Error Rate for 100 external samples

As can be seen from the above for a threshold value of 2, the set-up allows the co-operative and hostile intruders equally to access the system.

5. Conclusions and Future Work

5.1 Conclusions

A fingerprint based secure Bluetooth device communication was carried out using COTS software and Verifinger hardware. The system successfully compressed and encrypted the data essential for Bluetooth Protocol communication. In order to evaluate the performance of the system for co-operative users and hostile intruders experiments were carried out. Analysis of these experiments led to the study of the various constraints which were affecting the results.

During the study, various possibilities of converting the unique biometric data into a feature vector were considered. Based on various considerations a scheme was devised to compress the pixel data from a 260x300 (i.e. 78,000 pixels) fingerprint image to 128 bits. This meant 312kbytes (78,000 pixels x 4 bits/pixel) were compressed to 128 bits using rotation and translation invariant feature set which we propose in this thesis.

From the analysis of the data obtained from the results, the following conclusions were drawn:

- For a threshold value of 1.05, the system had an equal error rate for FAR and FRR.

- The fingerprints with greater no of extractable features have a better chance of being correctly recognized (Fig 3.7).
- The fingerprints with lesser number of distinguished features have a higher rate of being recognized as a correct fingerprint.
- The uniqueness of the fingerprint i.e., "non-identicality" with other fingerprints may generate incorrect statistics.
- In cases where the data of one of the features, matches the data of another feature, of another fingerprint, it gets recognized as the other fingerprint.
- The inclusion of 100 external samples, representing hostile intruders, also did not affect the stability of the system. The system had an Equal Error Rate of 2 for external samples which was a low error rate as per a stable system.

Our suggested approach provides a unique approach where the Biometrics data is not appended into the Bluetooth payload. Instead, it is utilized for pairing the Bluetooth devices and hence has the advantages over other systems where the data field is appended into the payload (as shown in Fig 1.1). For pairing between two devices a Bluetooth passkey or a PIN is entered into both the devices.

5.2 Future Work

The above data may provide a useful guide when attempting to use COTS software and Verifinger hardware to build a Bluetooth system utilizing the biometrics data for communication. As the results showed, increasing the number of people didn't destabilize the system. However if the system needs to be scaled upwards by adding more users, the number of parameters for studying the fingerprints can be increased. One such example that could have been incorporated was the standard deviation of the fingerprint features. Data field of the initialization key has 128 bits and there is enough space left in this data field from the data generated by the proposed scheme.

In order to improve and avoid the erroneous results generated by similar fingerprint samples like template 6 & template 7, a weightage factor α can be considered to assign more weightage to certain parameters, which provide more information with regards to the fingerprints.

Just like the mean value of the various parameters discussed in section 3.4, Standard Deviation can also be used as a part of the scheme. Similarly the orientation of the farthest ridges (and bifurcations) with the mean ridge (and bifurcation, respectively) can be few suggested parameters which could have given more stability.

Other biometrics features can be combined with fingerprints to provide a more robust authentication and verification system.

References:

[Anthony 04] Adam Anthony –"An Investigation of Remote Authentication Schemes:

The Key Scan Project"- http://www.wooster.edu/cs/seniors/2004/adamAnthony.html

[Jain 04] Jain, A.K.; Ross, A.; Pankanti, S.;- "Biometrics: a tool for information security."- **"**Information Forensics and Security, IEEE Transactions on" Volume 1, Issue 2, June 2006 Page(s):125 - 143

[An 04] Anil K.Jain –"Biometric Recognition: How Do I Know Who You Are?"- Signal Processing and Communications Applications Conference, 2004. Proceedings of the IEEE 12th Publication Date: 28-30 April 2004, page(s): 3- 5

[Anil Jain 04] Anil K.Jain, Arun Ross and Salil Prabhakar-"an Introduction to Biometric Recognition"- Circuits and Systems for Video Technology, IEEE Transactions on

Volume 14, Issue 1, Jan. 2004 Page(s):4 - 20

[Biometric Assessment] Biometric Technical Assessment

http://bio-tech-inc.com/Bio_Tech_Assessment.html /

[Biometric Report]Biometric Product Testing Final Report - National Physical Laboratory - http://www.cesg.gov.uk/site/ast/biometrics/media/BiometricTestReportpt1.pdf

[Bluetooth_Specification] "The Bluetooth Core Specification, v.1.2", November 2003. http://www.bluetooth.org/spec/

[Cherukuri 2003] Sriram Cherukuri, Krishna K Venkatasubramanian, Sandeep K S Gupta, " BioSec: A Biometric Based Approach for Securing Communication in Wireless Networks of Biosensors Implanted in the Human Body" -2003 International Conference on Parallel Processing Workshops (ICPPW'03) p. 432

[Clancy 2003] T.Charles Clancy, Negar Kiyavash and Dennis J.Lin, Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications, Berkley, California November 2003 Page(s): 45 - 52

[David 2003] M. **W.** David, *G.* **A.** Hussein, K. **Sakurai** –" Secure Identity Authentication and Logical Access Control for Airport Information Systems" Security Technology, 2003. Proceedings. IEEE 37th Annual 2003 International Carnahan Conference on Volume , Issue , 14-16 Oct. 2003 Page(s): 314 – 320

[Dellutri 2005]Fabio Dellutri, Gianluigi Me, Maurizio A. Strangio –"Local Authentication with Bluetooth enabled Mobile Devices"- Proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services" 23-28 Oct.2005 Page(s):72-72

[Gupta 2005] Gupta, P.; Ravi, S.; Raghunathan, A.; Jha, N.K.-" Efficient fingerprint-based user authentication for embedded systems" Design Automation Conference, 2005. Proceedings. 42nd 13-17 June 2005 Page(s):244 - 247

[Hahnsang 2005] Hahnsang Kim; Dabbous, W.; Afifi, H.- " A bypassing security model for anonymous Bluetooth peers" Wireless Networks, Communications and Mobile Computing, 2005 International Conference on ,Volume 1, 13-16 June 2005 Page(s):310 - 315 vol.1

[Jain 2003] Vivek Jain and Ramesh C. Joshi –"Integrating Bluetooth, Biometrics and Smartcards for Personal Identification and Verification" Proceedings of National Symposium on Emerging Trends In Networking and Mobile Communication.

[Woodward 2003]John D. Woodward, Nicholas Orlans, and Peter Higgins. Identity Assurance in the Information Age: Biometrics. McGraw-Hill/ Osbourne, New York, 2003. 20, 47, 51,52

[Li 2005] Li, H.; Mukesh Singhal-"A key establishment protocol for Bluetooth scatternets"Distributed Computing Systems Workshops, 2005. 25th IEEE International Conference on 6-10 June 2005 Page(s):610 - 616

[MD_5]The MD5 Homepage.

[Ortega 04]Ortega-Garcia J., BigunJ., ReynoldsD., Gonzalez-Rodriguez,J. – "Authentication gets Personal with Biometrics" -*Signal Processing Magazine, IEEE ,* Volume 21, Issue 2 pg 50- 62, March 2004.

[Patricia 2004] Patricia McDermott-Wells -"What is Bluetooth?" This paper appears in: "**Potentials,IEEE**", Date:Dec.2004-Jan.2005Volume:23,Issue:5 page(s):33-35

[Poon 2006]Poon, C.C.Y. Yuan-Ting Zhang Shu-Di Bao - Chinese Univ. of Hong Kong, Shatin, China: " Communications Magazine, IEEE" Date: April 2006 Volume: 44, Issue: 4 page(s): 73- 81

[Meenen 2005] Ray, M.; Meenen, P.; Adhami, R.;"A novel approach to fingerprint pore extraction" System Theory, 2005.SSST '05. Proceedings of the Thirty-Seventh Southeastern Symposium on 20-22 March 2005 Page:282 - 286

[Reillo 2004]Sanchez-Reillo, R.; Liu-Jimenez, J.; Entrena, L.; Garcia-Lorenz, M.-"Anti-trojan security module for biometric authentication tasks" Security

Technology, 2004. 38th Annual 2004 International Carnahan Conference on 11-14 Oct. 2004 Page:140 – 144

[Shen 99] Weicheng Shen; Tieniu Tan-"Automated Biometrics_Based Personal Identification" Proceedings of the National Academy of Sciences of the United States of America, 1999 Volume 96, Issue 20, pp. 11065-11066

[Tikkanen 03] Pauli Tikkanen, Seppo Puolitaival, Ilkka Känsälä "Capabilities of Biometrics for Authentication in Wireless Devices" - Audio- and Video-Based Biometric Person Authentication" Volume 2688/2003 page 796-804.

[Limt 2001]Yujin Limt, Jesung Kim, Sang Lyul Min, and Joong So0 Mat - "Performance Evaluation of the Bluetooth-based Public Internet Access Point" Information Networking, 2001. Proceedings. 15th International Conference on 31 Jan.-2 Feb. 2001 Page:643 - 648

1 Biometric Technical Assessment - http://bio-tech-inc.com/Bio_Tech_Assessment.html

2 Biometric Product Testing Final Report - National Physical Laboratory - http://www.cesg.gov.uk/site/ast/biometrics/media/BiometricTestReportpt1.pdf

[Yaniv 05] Yaniv Shaked and Avishai Wool-" Cracking the Bluetooth PIN" International Conference On Mobile Systems, Applications And Services Proceedings of the 3rd international conference on Mobile systems, applications, and services Seattle, Washington SESSION: Shake 'em, but don't crack 'em 2005 Pages: 39 - 50

APPENDIX

```
#include <stdio.h>
#include <string.h>
#include <iostream>
#include <limits.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <string>
#include <cmath>



float  TwoDimensionDistance(int *x1,int *x2,int*y1,int*y2);
int     readInputDataFile(int *,int *);
void   readfile(char *,int *, int *,int);
void   allDetailsAboutOnePerson(float,float,float,int);
void   callFinalProcessingFunction();
void    buildConcatenatedStringfromMatrix(char [][18]);

static float distanceArray[1000] = {0};
static float distanceBArray[1000] = {0};
static float distanceRArray[1000] = {0};
static float calculateMaxDistanceValue();
static float calculateMaxDistance();
static float calculateMaxRDistanceValue();
static float calculateMaxRfileDistanceValue();
static float calculateMaxBfileDistanceValue();
static float calculateMaxBDistanceValue();
static float calculateMaxFile();
static  int g_noOfRows = 0;
static  int g_input_noOfRows = 0;
static   int x_Rarray[1000]= {0};
static   int y_Rarray[1000] = {0};
static  int x_Barray[1000]= {0};
static   int y_Barray[1000] = {0};
static  int x_array[1000]={0};
static  int y_array[1000]={0};
static  int x_input_array[100]={0};
static  int y_input_array[100]={0};
static  int x_input_Barray[100]={0};
static   int x_input_Rarray[1000]= {0};
static   int y_input_Rarray[1000]= {0};
static   int y_input_Barray[1000]= {0};
static int B=0;
static int R=0;
static int Binput=0;
static int Rinput=0;
static   int noOfFiles = 0;
static   int minutiae=0;
static int gTotalR = 0;
static int gTotalB = 0;
static int MeanaveMinutiae = 0;
static int total=0;
static int yes=0;
static int reject=0;
static int TAT=0;
static float totalMaxDistanceValue  = 0.0;
static   float maxdistanceforridges = 0.0;
static   float maxdistanceforBifurcations = 0.0;
static  float thresh=0.0;
float X_I[6] = {0.0};
float XIARRAY[110][6] = {0.0};
float T_IJ[11][6] = {0.0};

static int count_xi = 0;
int main()
{
  /* reading input data file */
        int i = 0;
        int Mean_of_number_of_ridges = 0;
        int Mean_of_number_of_bifurcations = 0;
        char concatStringMatrix[11][18] = {0};

        noOfFiles =
readInputDataFile(&Mean_of_number_of_ridges,&Mean_of_number_of_bifurcat
ions);
        printf("Mean_of_number_of_ridges Value
:%d\n",Mean_of_number_of_ridges);
        printf("Mean_of_number_of_bifurcations Value
:%d\n",Mean_of_number_of_bifurcations);

        printf("T IJ is :\n");



        int k = 0;int l = 0;

        for (k = 0; k<11; k++)
        {
                for(l = 0 ; l < 6; l++)
                {
                        printf("%f ",T_IJ[k][l]);
                }printf("\n");l = 0;
        }


        buildConcatenatedStringfromMatrix(concatStringMatrix);
        getchar();

        char *fileNameBegining[11] =
{"pun","yon","drpandya","s","v","su","sid","pit","nit","henry","sri"};
        int fCount =0;
        char filePath[20] = "C:\\Temp\\";
        char buffer[5];
        char fileToRead[150]= {0};



        i++;
        while ( i < 11 || fCount <10)
        {
                if (fCount == 11) break;
                if (i == 11){ i = 1; fCount++;}
                itoa(i,buffer,10);
                strcpy(fileToRead,filePath);
                strcat(fileToRead,fileNameBegining[fCount]);
                strcat(fileToRead,buffer);
                strcat(fileToRead,".fp");

                //printf("i = %d,filename is :%s\n",i ,fileToRead);
                i++;memset(buffer,0,5);

                readfile(fileToRead,&Rinput,&Binput,fCount);

//      printf("\nthe maximum distance for ridges in file is:%f\n",FileRMax);
//      printf("\nthe maximum distance for bifurcations in file
is:%f\n",FileBMax);


        }

        callFinalProcessingFunction();

        return 0;
```

```
}

void buildConcatenatedStringfromMatrix(char stringMatrix[][18])
{
        int i = 0;
        int j = 0;
        char temp[3] = {0};

        for(i = 0; i < 11; i++)
        {
                for(j = 0; j < 6; j++)
                {
                        if (T_IJ[i][j]<99)
                        {
                                _itoa(int(T_IJ[i][j]),temp,10);
                        }
                        else
                        {
                                _itoa(int(T_IJ[i][j])/10,temp,10);
                        }
                        strcat(stringMatrix[i],temp);
                        memset(temp,0,3);
                }printf("\n");printf("stringMatrix[%d]\t%s",i,stringMatrix[i]);
        }
}
void allDetailsAboutOnePerson(float totalMaxDistanceValue,float
maxdistanceforridges,float maxdistanceforBifurcations,int noOfFiles)
{
        float Meanofmaxdistanceforminutiae = 0;
        float Meanofmaxdistanceforridges = 0.0;
        float MeanofmaxdistanceforBifurcations = 0.0;

        Meanofmaxdistanceforminutiae = totalMaxDistanceValue/noOfFiles;
        Meanofmaxdistanceforridges = maxdistanceforridges/noOfFiles;
        MeanofmaxdistanceforBifurcations =
maxdistanceforBifurcations/noOfFiles;

        printf("maxdistanceforridges:%f\n",maxdistanceforridges);

        printf("maxdistanceforBifurcations :%f\n",maxdistanceforBifurcations);

        printf("MeanaveMinutiae Value :%d\n",MeanaveMinutiae);
        printf("Meanofmaxdistanceforminutiae
:%d\n",Meanofmaxdistanceforminutiae);
        printf("Mean of max distance for ridges Value
:%d\n",Meanofmaxdistanceforridges);
```

```
	printf("Mean of max distance for Bifurcations Value
:%d\n",MeanofmaxdistanceforBifurcations);


}
int	readInputDataFile(int *Mean_of_number_of_ridges,int
*Mean_of_number_of_bifurcations)
{
	FILE * infptr;
	char *fileNameBegining[11] =
{"pun","yon","drpandya","s","v","su","sid","pit","nit","henry","sri"};


	int noOfRows = 0;
	int x =0, y = 0, z =0,p = 0,
xRigdes=0,yRigdes=0,xBifurcations=0,yBifurcations=0;
	int count = 0;
//	int i;
	int angle=0;
	unsigned long int SumX=0;
	unsigned long int SumY=0;
	char t;

	float TwoDimensionMax=0;
	int fileCount =0;
	char filePath[20] = "C:\\Temp\\";
	char buffer[5];
	char fileToRead[150] = {0};

	float totalDistanceArrayValue = 0.0;
	float totalDistanceRArrayValue = 0.0;
	float totalDistanceBArrayValue = 0.0;
	float TwoDimensionBMax=0.0;
	float TwoDimensionRMax=0.0;

	int i = 1;
	while (i<=11 && fileCount < 11 )
	{
		if (i == 11)
		{
			i = 1;
		}

		g_noOfRows = 0;
		itoa(i,buffer,10);
		strcpy(fileToRead,filePath);
```

```
        strcat(fileToRead,fileNameBegining[fileCount]);
        strcat(fileToRead,buffer);
        strcat(fileToRead,".fp");
        printf("i = %d,filename is :%s\n",i , fileToRead);

        if ((infptr = fopen(fileToRead, "r")) == NULL)
        {
            printf("file empty\n");
            //exit(1);
            goto Process;
        }
        else
        {
          fscanf(infptr,"%d",&noOfRows);i++;
            //printf("no of runs=%d\n",noOfRows);
            while (noOfRows!=0)
            {
                fscanf(infptr,"%d  %d  %d  %c  %d", &x, &y, &z,
&t, &p);
                noOfRows--;
                x_array[count] = x;
                y_array[count] = y;
                //        x=0;y=0;
                count++;
                angle+= z;
                if (t=='R')
                {
                    x_Rarray[R] = x;
                    y_Rarray[R] = y;
                    //printf("x=%d\ty=%d\n",x,y);
                    xRigdes=xRigdes+ x_Rarray[R];
                    yRigdes=yRigdes+ y_Rarray[R];
                    //float TwoDimensionRMax=0;
                    //TwoDimensionRMax=
calculateMaxRDistanceValue();
                    //distanceRArray[fileCount] =
TwoDimensionRMax;
                    //TwoDimensionRMax=0.0;
                    x=0;y=0;        R++;
                }
                else
                {

                    x_Barray[B] = x;
                    y_Barray[B] = y;
                    //printf("x=%d\ty=%d\n",x,y);
```

```
                                        xBifurcations=xBifurcations+ x_Barray[B];
                                        yBifurcations=yBifurcations+ y_Barray[B];
                                        //float TwoDimensionBMax=0;
                                        //TwoDimensionBMax=
calculateMaxBDistanceValue();
                                        //distanceBArray[fileCount] =
TwoDimensionBMax;


                                        //TwoDimensionBMax=0.0;
                                        x = 0; y = 0;   B++;

                                }
                        }//while
                //      printf("no of runs=%d\n",count);
                }//if
                fclose(infptr);
                memset(buffer,0,5);
                g_noOfRows = count;

                TwoDimensionMax= calculateMaxDistanceValue();
                TwoDimensionRMax= calculateMaxRDistanceValue();
                TwoDimensionBMax= calculateMaxBDistanceValue();

                printf("TwoDimensionMax:%f\n",TwoDimensionMax);
                printf("TwoDimensionRMax:%f\n",TwoDimensionRMax);
                printf("TwoDimensionBMax:%f\n",TwoDimensionBMax);

                distanceArray[fileCount] = TwoDimensionMax;
                distanceBArray[fileCount] = TwoDimensionBMax;
                distanceRArray[fileCount] = TwoDimensionRMax;

                totalDistanceRArrayValue  += TwoDimensionRMax;
                totalDistanceBArrayValue  += TwoDimensionBMax;
                totalDistanceArrayValue   += TwoDimensionMax;

                printf("totalDistanceArrayValue =
%f\n",totalDistanceArrayValue);
                printf("totalDistanceRArrayValue =
%f\n",totalDistanceRArrayValue);
                //getchar();

                TwoDimensionMax=0.0;
                TwoDimensionRMax=0.0;
                TwoDimensionBMax=0.0;

                //fileCount++;
                memset(x_array,0,1000);
                memset(y_array,0,1000);
                memset(x_Barray,0,1000);
                memset(y_Barray,0,1000);
                memset(x_Rarray,0,1000);
                memset(y_Rarray,0,1000);
                count = 0;
                gTotalR += R;
                gTotalB += B;

                R=0;B=0;
Process:
        if ( i == 11)
        {
                /*int noOfFiles = 10;

                for (i = 0;i < noOfFiles; i++)
                {

                        printf("max distanceArray[%d] :%f\n",i,distanceArray[i]);
                        totalMaxDistanceValue += distanceArray[i];

                        //printf("max distanceRArray[%d]
:%f\n",i,distanceRArray[i]);
                        //maxdistanceforridges += distanceRArray[i];

                        //printf("max distanceBArray[%d]
:%f\n",i,distanceBArray[i]);
                        //maxdistanceforBifurcations += distanceBArray[i];
                }*/

                printf("no of bifurcations=%d\n",gTotalB);
                printf("no of ridges      =%d\n",gTotalR);

                minutiae= gTotalB+gTotalR;
                MeanaveMinutiae+=minutiae/10;
                *Mean_of_number_of_ridges  += gTotalR/10;
                *Mean_of_number_of_bifurcations += gTotalB/10;

                printf("totalDistanceArrayValue=%f\n",totalDistanceArrayValue);

        printf("totalDistanceRArrayValue=%f\n",totalDistanceRArrayValue);

                //getchar();
                int j = 0;
        //allDetailsAboutOnePerson(totalMaxDistanceValue,maxdistanceforridges
,maxdistanceforBifurcations,noOfFiles);
                T_IJ[fileCount][j] = minutiae/10;
                T_IJ[fileCount][j+1] = gTotalR/10;
                T_IJ[fileCount][j+2] = gTotalB/10;
                T_IJ[fileCount][j+3] = int(totalDistanceArrayValue/10);
                T_IJ[fileCount][j+4] = int(totalDistanceRArrayValue/10);
                T_IJ[fileCount][j+5] = int(totalDistanceBArrayValue/10);

                if (gTotalB%10 >5)*Mean_of_number_of_bifurcations +=1;
                if (gTotalR%10 >5)*Mean_of_number_of_ridges +=1;

                memset(fileToRead,0,100);

                MeanaveMinutiae =0;
                gTotalB = 0;
                gTotalR = 0;
                totalDistanceArrayValue = 0.0;
                totalDistanceRArrayValue = 0.0;
                totalDistanceBArrayValue = 0.0;
                fileCount++;
        }
  }//WHILE
  return fileCount;
}

float TwoDimensionDistance(int *x1,int *x2,int*y1,int*y2)
{

        int d1,d2;

        d1=(*x2>=*x1)?(*x2-*x1):(*x1-*x2);
        d2=(*y2>=*y1)?(*y2-*y1):(*y1-*y2);

        long int a=pow(d1,2);
        long int b=pow(d2,2);
        float k= a+b;
        float distanceBetweenPoints=sqrt(k);

        //printf("\nThe distanceBetweenPoints is:%f\n",distanceBetweenPoints);

        return sqrt(k);

}
```

```
float calculateMaxDistanceValue()
{
        float max_value= 0.0;
        //int i = 0,j= 0;
        float twoDimensionDistance = 0.0;

        for ( int s = 0;s <=g_noOfRows;s++)//loop for increasing the count ofthe
minutiae
        {

                for ( int t = s; t <=g_noOfRows;t++)
                {
                        twoDimensionDistance =
TwoDimensionDistance(&x_array[s],&x_array[t],&y_array[s],&y_array[t]);
                        //printf("\nThe twoDimensionDistance = %f max_value =
%f\n",twoDimensionDistance,max_value);
                        if(max_value<twoDimensionDistance)
                        {

        max_value=TwoDimensionDistance(&x_array[s],&x_array[t],&y_array[s
],&y_array[t]);
                        }

                }
        }
        //printf("\nThe max Distance is:%f\n",max_value);
        //printf("mean angle is :%f",Theta);
        return max_value;

}

float calculateMaxRDistanceValue()
{
        float max_value= 0.0;
        //int i = 0,j= 0;

        for (int s = 0;s <=R;s++)//loop for increasing the count ofthe minutiae
        {

        for ( int t = s; t <=R;t++)
                {

        if(max_value<TwoDimensionDistance(&x_Rarray[s],&x_Rarray[t],&y_R
array[s],&y_Rarray[t]))
                        {
```

```
		max_value=TwoDimensionDistance(&x_Rarray[s],&x_Rarray[t],&y_Rar
ray[s],&y_Rarray[t]);
				}

			}
		}
		return max_value;

}

float calculateMaxBDistanceValue()
{
float max_value= 0.0;
		//int i = 0,j= 0;

		for (int s = 0;s <=B;s++)//loop for increasing the count ofthe minutiae
		{

		for ( int t = s; t <=B;t++)
			{

		if(max_value<TwoDimensionDistance(&x_Barray[s],&x_Barray[t],&y_B
array[s],&y_Barray[t]))
				{

		max_value=TwoDimensionDistance(&x_Barray[s],&x_Barray[t],&y_Bar
ray[s],&y_Barray[t]);
				}

			}
		}

		return max_value;

}

void readfile(char * fileToRead, int * Rinput, int * Binput ,int fileIndex)
{
		FILE  * infponter;
	char  in_file[100] ={0};// "C:\\Temp\\yong2.fp";
		int noOfRows = 0;
		int x =0, y = 0, z =0,p = 0,
xRigdes=0,yRigdes=0,xBifurcations=0,yBifurcations=0;
		int count_read = 0;
		int i;
```

```
         int angle=0;

         //float Rangle=0,Bangle=0;
         //unsigned long int SumX=0;
         //unsigned long int SumY=0;
         char t;
         strcpy(in_file,fileToRead);
         if ((infponter = fopen(in_file, "r")) == NULL)
         {


                  printf("file empty\n");
                  exit(1);
         }
   else{
                     fscanf(infponter,"%d",&noOfRows);
                           printf("no of runs=%d\n",noOfRows);
                           while (noOfRows!=0)
                           {
                                             fscanf(infponter,"%d  %d  %d  %c  %d",
&x, &y, &z, &t, &p);
                                    //printf("TYPE: %c x :%d y: %d z: %d   \n
",t,x,y,z);

                                    noOfRows--;
                                    x_input_array[count_read] = x;
                                    y_input_array[count_read] = y;
                                    count_read++;
                                    angle+= z;
                                    if (t=='R')
                                    {
                                    x_input_Rarray[R] = x;
                                    y_input_Rarray[R] = y;
                                    x = 0; y = 0;
                                    xRigdes=xRigdes+ x_input_Rarray[R];
                                    yRigdes=yRigdes+ y_input_Rarray[R];
                                    R++;
                                    }
                                    else {
                                    x_input_Barray[B] = x;
                                    y_input_Barray[B] = y;
                                    x = 0; y = 0;
                                    xBifurcations=xBifurcations+ x_input_Barray[B];
                                    yBifurcations=yBifurcations+ y_input_Barray[B];
                                    B++;

                                    }
```

```
                                //x = 0; y = 0;
                        }

                //      printf("no of runs=%d\n",count);
                }
                fclose(infponter);
        //int k = 0;
        int j=0;*Rinput=R; *Binput=B;
/*      printf("For the Bifurcations:");
        printf("\nX:\tY:\t\n");
        for (;k < *Binput; k++)
        {
                printf("%d\t%d\n",x_input_Barray[k],y_input_Barray[k]);
        }

        printf("For the Ridges:\n");
        printf("X:\tY:\t\n");
        for (;j < *Rinput; j++)
        {
                printf("%d\t%d\n",x_input_Rarray[j],y_input_Rarray[j]);
        }*/
i = 0;
        /*printf("for the fingerprint:\n");
        printf("X:\tY:\t\n");
        for (;i < count_read; i++)
        {
                printf("%d\t%d\n",x_input_array[i],y_input_array[i]);
                //SumX=SumX+ x_array[i];
                //SumY=SumY+ y_array[i];
        }*/
                total=R+B;
                printf("\ntotal minutiae=%d",total);
                g_input_noOfRows = count_read;
                float FileMax=0;
                float FileRMax=0;
                float FileBMax=0;
                FileMax  = calculateMaxFile();
                FileRMax = calculateMaxRfileDistanceValue();
                FileBMax = calculateMaxBfileDistanceValue();

                int h=0;

        //      getchar();


        //      int h=0;
```

```
                        X_I[h] = total;
                X_I[h+1] = R;
                X_I[h+2] = B;
                X_I[h+3] = FileMax;
                X_I[h+4] = FileRMax;
                X_I[h+5] = FileBMax;

                        for(int q=0;q<6;q++)
                {
                        XIARRAY[count_xi][q] = X_I[q];
                }
                        for(q=0;q<6;q++)
                {
                        printf("\n%f",X_I[q]);
                }

                total = 0;R = 0; B = 0;
                count_xi++;

                printf("\nCount XI is : %d\n",count_xi);

}


float calculateMaxFile()
{
        float max_value= 0.0;
        int i = 0,j= 0;

        for (int s = 0;s <g_input_noOfRows;s++)//loop for increasing the count
ofthe minutiae
        {

        for ( int t = s; t <g_input_noOfRows;t++)
                {

        if(max_value<TwoDimensionDistance(&x_input_array[s],&x_input_arra
y[t],&y_input_array[s],&y_input_array[t]))
                        {

        max_value=TwoDimensionDistance(&x_input_array[s],&x_input_array[t
],&y_input_array[s],&y_input_array[t]);
                        }
                        }
                }
//              printf("\nThe max Distance is try debugging it:%f\n",max_value);
                //printf("mean angle is :%f",Theta);
                return max_value;

}

float calculateMaxRfileDistanceValue()
{
                float max_value= 0.0;
                //int i = 0,j= 0;

                for (int s = 0;s <R;s++)//loop for increasing the count ofthe minutiae
                {

                for ( int t = s; t <R;t++)
                        {

                if(max_value<TwoDimensionDistance(&x_input_Rarray[s],&x_input_Ra
rray[t],&y_input_Rarray[s],&y_input_Rarray[t]))
                                {

                max_value=TwoDimensionDistance(&x_input_Rarray[s],&x_input_Rarra
y[t],&y_input_Rarray[s],&y_input_Rarray[t]);
                                }


                        }
                }
                return max_value;

}

float calculateMaxBfileDistanceValue()
{
                float max_value= 0.0;
                //int i = 0,j= 0;

                for (int s = 0;s <B;s++)//loop for increasing the count ofthe minutiae
                {

                for ( int t = s; t <B;t++)
                        {
```

```
        if(max_value<TwoDimensionDistance(&x_input_Barray[s],&x_input_Ba
rray[t],&y_input_Barray[s],&y_input_Barray[t]))
                        {

        max_value=TwoDimensionDistance(&x_input_Barray[s],&x_input_Barra
y[t],&y_input_Barray[s],&y_input_Barray[t]);
                        }

                }
        }
        return max_value;
}
void callFinalProcessingFunction()
{
        int speciGroup = 0;
        int speciGroupFile = 0;
        int t_ij_ValueIndex = 0;
        int b = 0;
        int i = 0;
        int j = 0;
        float fSample[10]={0.0};
        float fValue[110*11]   = {0.0};
        float fValue_comp[110] =      {0.0};
        float FsquareVa[110] ={0.0};
        float Real =0.0;
        float Deal =0.0;


        float X = 0.0;
        float T = 0.0;
        float V =0.0;
        FILE *fPtr = NULL;

        /*printf("Enter which specimen group to select( chaose 1 to 11):\n");
        scanf("%d",&speciGroup);
        printf("Enter which specimen group file to select( chaose 1 to 10):\n");
        scanf("%d",&speciGroupFile);
        printf("Enter which T IJ Index to select( chaose 1 to 11):\n");
        scanf("%d",&t_ij_ValueIndex);*/

        fPtr = fopen("answers11.txt","w");
        if (!fPtr)
        {
                exit(1);
        }
```

```
/*for ( i = 0 ; i < 6; i++)
{
        X = XIARRAY[(speciGroup-1*10)+(speciGroupFile-1)][i];
        T = T_IJ[t_ij_ValueIndex-1][i];
        fValue += (X - T)/T;
}
printf("Final Value %f\t",fValue);*/
int fVal = 0;
int fVal_comp=0;
/*for ( i =0 ; i < 110; i++)
{
        for (j = 0; j < 6; j++)
        {
        printf("%f ",XIARRAY[i][j]);
        }printf("\n");
}*/
                for ( int k =0 ; k < 11; k++)
{

 for (i=100;i<110;i++)
 {



                for(j = 0; j < 6; j++)
                {
                        X = XIARRAY[i][j];
                        T = T_IJ[k][j];
                        V =(X - T)/T;

        //      fprintf(fPtr,"the real value is%f\n",V);
                Real= fabs(V);
        //              fprintf(fPtr,"the absolute value is%f\n",Real);
        //printf("Real:%f\n",Real);
                fValue[fVal] += Real;
                }fVal += 1;


 }
}
printf("fVal:%d\n",fVal);
fprintf(fPtr," the value of the template is:\n");
for ( i = 100 ; i < 110; i++)
{
        fSample[b]=fValue[i];
        b++;
    }
    for ( b = 0 ; b < 10; b++)
    {

        fprintf(fPtr,"\nthe value of the copied array is%f\n",fSample[b]);


    }
    //  fclose(fPtr);
    //b=0;
    fprintf(fPtr," the value of the data is:\n");

        for (int  q= 0 ; q < 10; q++)
        {


                for(int h=q;h<110;h=h+10)
                {
                    if (fSample[q]<=fValue[h])
                    {yes++;}


                    else{fprintf(fPtr,"The fingerprint %d is
being recognised as template :%d\n",q,h/10);reject++;break;}
                    reject=0;
                }




            fprintf(fPtr,"The value of the accepted %d\n",yes);
        /*  fprintf(fPtr,"The value of the rejected is%d\n",reject);
            fprintf(fPtr,"The threshold value is:%f\n",thresh);
        */  yes=0;

        }


     fclose(fPtr);
     }
```