CERIAS Tech Report 2010-10

An Evaluation of Template Splitting to Prevent Sample Reconstruction from Fingerprint Templates

by Ashwin Mohan
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Ashwin Mohan

Entitled An Evaluation of Template Splitting to Prevent Sample Reconstruction from Fingerprint Templates

For the degree of Master of Science

Is approved by the final examining committee:

Stephen J. Elliott, Ph.D.
Chair

Shimon K. Modi, Ph.D.

Elisa Bertino, Ph.D.

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Stephen J. Elliott, Ph.D.

Approved by: Eugene H. Spafford, Ph.D. 07/08/2010
Head of the Graduate Program Date

Graduate School Form 20
(Revised 1/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

An Evaluation of Template Splitting to Prevent Sample Reconstruction from Fingerprint Templates

For the degree of Master of Science

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Teaching, Research, and Outreach Policy on Research Misconduct (VIII.3.1),* October 1, 2008.*

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Ashwin Mohan
Printed Name and Signature of Candidate

07/09/2010
Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/viii_3_1.html

AN EVALUATION OF TEMPLATE SPLITTING TO PREVENT SAMPLE RECONSTRUCTION FROM FINGERPRINT TEMPLATES

A Thesis

Submitted to the Faculty

of

Purdue University

by

Ashwin Mohan

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

August, 2010

Purdue University

West Lafayette, Indiana

To my parents for their constant support and encouragement

ACKNOWLEDGMENTS

The author would like to thank Dr. Stephen Elliott, Dr. Shimon Modi and Dr. Elisa Bertino at Purdue University for their continued advice and guidance during the course of this thesis.

The author would also like to thank Benny Senjaya at Purdue University and Dr. Feng at Michigan State University for the help provided during the course of this project.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

Table

LIST OF FIGURES

Page

Figure

LIST OF ABBREVIATIONS

AD - Auxiliary Data

ADA - American Dermatoglyphics Association

AFIS - Automated Fingerprint Recognition Systems

AHGBEA - Ad Hoc Group on Biometrics in E-Authentication

CCD - Charged Coupled Device

CMOS - Complementary Metal Oxide Semiconductor

FBI - Federal Bureau of Investigation

FM - Frequency Modulation

FNMR - False Non Match Rate

FTIR - Frustrated Total Internal Reflection

FVC - Fingerprint Verification Competition

IBG - International Biometrics Group

INCITS - InterNational Committee for Information Technology Standards

ISO - International Organization for Standardization

NIST - National Institute of Science and Technology

PI - Pseudo Identities

PIC - Pseudo Identity Comparator

PIE - Pseudo Identity Encoder

PIR - Pseudo Identity Recoder

PIV - Pseudo Identity Verifier

SD - Supplementary data

TURBINE - TrUsted Revocable Biometric IdeNtitiEs

ABSTRACT

Mohan, Ashwin. M.S., Purdue University, August 2010. An Evaluation of Template Splitting to Prevent Sample Reconstruction from Fingerprint Templates. Major Professor: Dr. Stephen Elliott.

Current research in fingerprint recognition systems have shown that given a fingerprint template, an approximation of the original fingerprint sample can be created. In this thesis, the capability of template splitting to prevent sample reconstruction from fingerprint templates is evaluated. An attack simulation was formulated as part of this thesis for testing template splitting within a fingerprint verification setup in its ability to prevent sample reconstruction. False Non Match Rate (FNMR) was used as the performance metric. Statistical analysis of the FNMR showed that the use of template splitting results in a significant decrease in the capability of approximate fingerprint samples to get matched within the fingerprint system.

CHAPTER 1. INTRODUCTION

A biometric trait is a characteristic that is unique to each individual. It can be physical characteristics like fingerprints and iris or behavioral characteristics like voice and signature. A system that uses biometric traits for automated recognition of an individual is a biometric system (International Organization for Standardization [ISO], 2008, p. 3). In a biometric system, a user's biometric trait is captured using a biometric capture device, whose main component is a biometric sensor. The raw data acquired from the biometric sensor is either digital or an analog representation of the biometric characteristic and is called a biometric sample (ISO, 2008, p. 5). Generally, the biometric sample contains more data than is required for biometric recognition. To remove any superfluous information that does not contribute to recognition, a salient set of features are extracted from the biometric sample. This set of features is stored within the biometric system and the stored set is referred to as a biometric template (ISO, 2008, p. 5). The biometric template is used for comparisons during the biometric recognition process in place of the biometric sample (ISO, 2008, p. 5).

In the creation of a biometric template, any biometric information that is not part of the extracted feature set but is part of the original biometric sample is lost. As a result, it's expected that if a biometric template is available, it cannot be used to recreate the biometric sample. Several biometric providers have claimed to this end by saying that the template generation process is a one-way function. Examples of these claims are given below:

1. I/O Software: "It [a biometric template] cannot be used to reconstruct an image to reveal a person's identity to someone else" (Hill, 2001, p. 36).

2. Veridicom: “However, you cannot recreate the original fingerprint from the minutiae [template] data” (Hill, 2001, p. 36).
3. Atmel: “The fingerprint cannot be re-constructed from the template” (Atmel, 2004, p. 38).
4. DigitalPersona: “These fingerprint templates are created whenever a user places a finger on the reader, and encoded with a one-way algorithm that cannot be reversed to recreate the scan of that fingerprint” (DigitalPersona, 2006, p. 14).
5. UPEK: “UPEK algorithms extract unique features to create a template and discard the fingerprint image to preserve privacy. It is not possible to recreate a fingerprint image from the template data” (UPEK, 2008, secure section, para. 2).

Unfortunately, these claims by biometric providers do not make the distinction between recreating the exact original biometric sample and creating a good likeness of the original biometric sample from the biometric template. The International Biometrics Group [IBG] (2002) made this distinction and acknowledged that while the biometric template couldn’t be used to recreate the exact original biometric sample (p. 23), creating an approximation of the original biometric sample was still a possibility (p. 29).

1.1. Statement of Problem

Current research in the fingerprint modality has developed methods by which a fingerprint template can be used to create a fingerprint sample that has a good likeliness to the original fingerprint. The intent of this study was to prevent the sample reconstruction from fingerprint templates.

1.2. Significance of Problem

A malicious individual can use the approximate fingerprint sample created from a user's complete fingerprint template in a masquerade attack on a fingerprint system (Hill, 2001). In the event of a successful masquerade attack on the fingerprint system, the malicious individual gets authorized access impersonating as the user whose fingerprint template was stolen and used in creating the approximate sample. If the imposter can accomplish multiple intrusions on the same system, or similar endeavors on other fingerprint systems where the user is enrolled, such behavior would amount to taking over the user's identity (Clarke, 2002, section 3.4, para. 1; Newman & McNally, 2005, p. 2).

There are three types of negative effects typically associated with identity theft for individuals as outlined by Newman and McNally (2005):

1. Financial loss: An attacker can gain monetary value by making use of the stolen identity to take over of existing bank accounts and credit cards for the user. A conservative estimate given by the National Fraud Center of total financial costs of identity theft to individuals was up to 50 billion dollars a year (p. 34).
2. Judicial and social implications: If an attacker uses the identity to perform illegal or criminal activity, the user can be charged for persecution and it would be hard to get a successful disposal of the case (p. vi). There can be a lot of delay between the theft of the identity and its detection, in which case the user may be unable to get a job or qualify for loans or any other types of social opportunities (p. 35).
3. Personal harm: The user would be forced to spend considerable time and effort in order to solve the issue besides suffering the feeling of violation and having the fear for individual safety. He would also be strained with the constant pestering by law enforcement and credit agencies and deal with hardships in relationships (p. 35). As an example, a victim of identity theft was told to travel to Florida to petition to its court in order to remove a fake account placed on his credit line in that State (p. 36).

1.3. Purpose of Study

There exists literature on how to prevent a masquerade attack on fingerprint systems that uses an approximate fingerprint sample. The author’s study added to this body of work.

Current methods to subvert a masquerade attack focus on the availability of the fingerprint template. These methods protect the fingerprint template at its storage location so that it does not get compromised. This research took a different approach by focusing on the reconstruction process. If an approximate sample is successfully created from a compromised fingerprint template, but the fingerprint system declares it to be invalid when presented for recognition, the masquerade attack is ineffective. This will happen when the approximate sample is not a good approximation of the original fingerprint sample.

This study evaluated whether a fingerprint template splitting scheme that splits a fingerprint template into two parts by dividing the set of constituent minutiae points along their y-coordinates could prevent a good approximation of the original fingerprint sample from being constructed by using the template. The fingerprint template splitting scheme was provided by Modi (personal communication, October 15, 2009). The ability of the fingerprint template splitting scheme to prevent a good reconstruction of an approximate fingerprint sample was measured in terms of False Non Match Rate (FNMR).

1.4. Definition of Terms

Biometric System

“System for the purpose of the automated recognition of individuals based on their behavioral and biological characteristics “ (ISO, 2008, p. 3).

Biometric Sample

“Analog or digital representation of biometric characteristics prior to biometric feature extraction and obtained from a biometric capture system “ (ISO, 2008, p. 5).

Biometric Template “Set of stored biometric features comparable directly to biometric features of a probe biometric sample” (ISO, 2008, p. 5).

Biometric Feature “Numbers or labels extracted from biometric samples and used for comparison” (ISO, 2008, p. 4).

Biometric Enrollment “The process of collecting a biometric sample(s) from an individual, and the subsequent construction and storage of a reference template(s) and associated data representing the individuals identity” (Ad Hoc Group on Biometrics in E-Authentication [AHGBEA], 2007, p. 31).

Biometric Verification “A one-to-one comparison of an individual’s biometric sample with a single biometric reference template in order to validate an explicit positive claim of identity” (AHGBEA, 2007, p. 32).

Biometric Identification “The one-to-many process of comparing a submitted biometric sample against all or a specified subset of the biometric reference templates on file to determine whether it matches any of the stored templates and, if so, the identity of the enrollee whose template was matched” (AHGBEA, 2007, p. 33).

False Match Rate (FMR) “[It] is the expected probability that a sample will be falsely declared to match a single randomly selected [genetically different] template” (Mansfield & Wayman, 2002, p. 5).

False Non Match Rate “[It] is the expected probability that a sample will be falsely declared not to match a template of the same measure from the same user supplying the sample” ” (Mansfield & Wayman, 2002, p. 5).

Biometric Performance “achievable recognition accuracy and speed, the resources required to achieve the desired recognition accuracy and speed as well as the operational and environmental factors that affect the accuracy and speed” (Jain, Ross, & Prabhakar, 2004, p. 4).

1.5. Assumptions

The assumptions for this study were:

1. The fingerprint database provided by Modi (personal communication, October 15, 2009) was representative of other fingerprint databases.
2. The steps for creating an approximate fingerprint sample as implemented by the author were consistent with those outlined in Feng and Jain (2009).
3. The software components Neurotechnology VeriFinger® 6.0 feature extractor and Neurotechnology VeriFinger® 6.0 matcher performed as given in their product specifications.
4. The fingerprint templates were stored in an unencrypted, uncompressed form within the fingerprint data storage subsystem.
5. The fingerprint templates consisted of minutiae information defined according to the standard provided in ISO (2005).
6. The FNMR was not be biased by the software components or the hardware setup for testing.

1.6. Delimitations

The delimitations for this study were:

1. The research was limited to the fingerprint modality.
2. Data collection was not conducted as part of the research process.
3. A single fingerprint database provided by Modi (personal communication, October 15, 2009) was used.

4. A single feature extractor and a single fingerprint feature matcher was used, Neurotechnology VeriFinger® 6.0 feature extractor and Neurotechnology VeriFinger® 6.0 matcher respectively.
5. A single fingerprint template splitting scheme provided by Modi (personal communication, December 16, 2009) was used.
6. Each individual fingerprint template was extracted using the Neurotechnology VeriFinger® 6.0 feature extractor from a single impression of the fingerprint sample.
7. The fingerprint templates extracted from the fingerprint samples consisted only of minutiae information.

CHAPTER 2. LITERATURE REVIEW

The aim of this chapter is to provide a basic understanding of the domain under consideration as well as motivation to take a new direction in research. The following review of the literature has six sections. The first section discusses the design, components and implementation architectures for biometric systems. The second section discusses the history of fingerprinting and the processes within a fingerprint recognition system. It also explores the composition and creation of fingerprint templates. The third section discusses the attacks on biometric systems. It investigates the different classes of attacks and describes the steps for a specific class of attacks that replays a previous biometric signature. It is here that the masquerade method and the hill climbing method used to construct a biometric sample from a biometric template are described. The fourth section discusses the feature transformation schemes and biometric cryptosystems as methods to prevent sample reconstruction by protecting biometric templates at storage. It also describes biometric template protection as part of the TrUsted Revocable Biometric IdeNtitiEs (TURBINE) project. The fifth section discusses existing work done for template splitting. The last section briefly evaluates error rates.

2.1. Overview of Biometric Systems

2.1.1. System Model

Mansfield and Wayman (2002) initially proposed the general biometric model. The model provides a visual representation of the components and process flow in the biometric system. The model was amended and included in ISO (2006). Fig 2.1 shows the updated model.

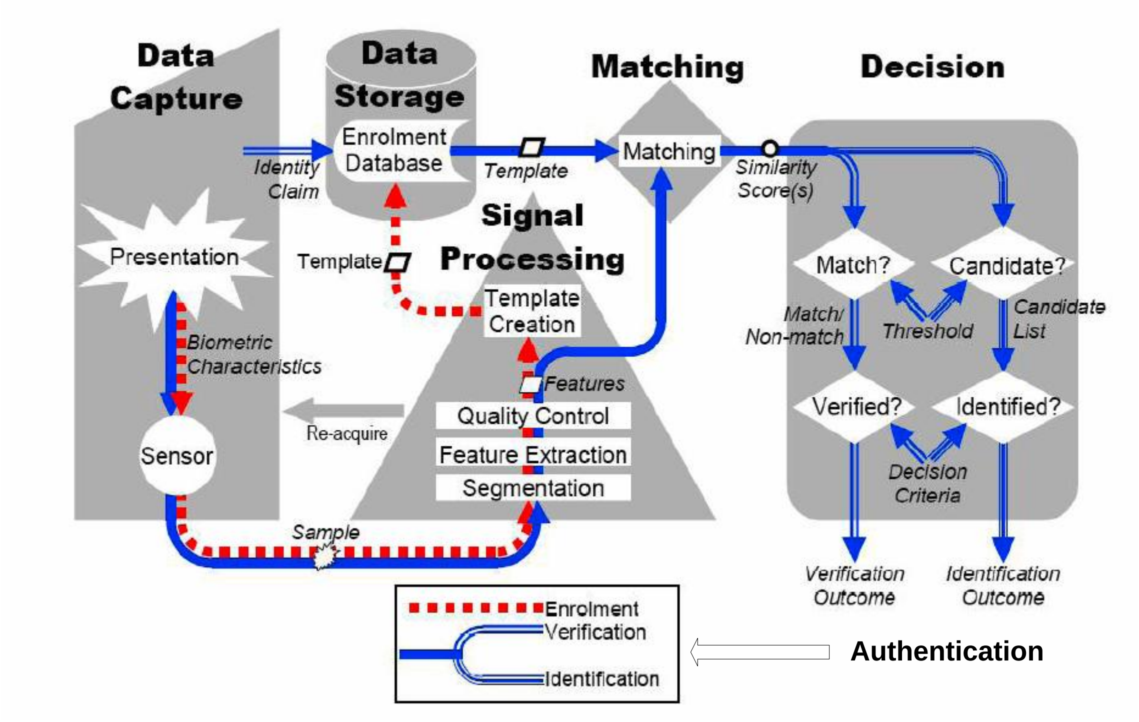

Figure 2.1 Updated General Biometric System Model (ISO, 2006)

2.1.1.1. Components

The updated general biometric model from ISO (2006) has the following components as shown in Fig 2.1: data capture, signal processing, data storage, matching, and decision. In the data capture subsystem the user presents his/her biometric trait to the sensor that acquires biometric sample(s). The signal processing subsystem takes the biometric sample(s) to create a biometric template /feature set through the process of segmentation, feature extraction and quality control. The storage subsystem manages the biometric template. The matching subsystem compares the features extracted from new biometric sample(s) to the template stored in the system and provides a similarity score, which indicates a closeness of fit. The decision subsystem uses the similarity score to give an outcome as a match or non-match for an authentication transaction that is based on the system decision policy.

2.1.1.2. Process Flow

The general biometric system model incorporates an authentication process. The authentication process consists of three subsystems: enrollment, verification and identification. These are shown with solid red and blue lines in Fig 2.1.

In enrollment, biometric sample(s) are collected from an individual through the biometric capture device. The samples(s) are processed through the signal processing subsystem and a reference biometric template is constructed. The reference template is subsequently stored in the enrollment database with an identifier for the individual that can be the name or any other such unique id.

Identification is a one-to-many process in which an individual submits biometric sample(s) and makes no or an implicit negative claim to an enrolled identity (AHGBEA, 2007, p. 32). The signal processing system takes the biometric sample(s) as input and provides a feature set. In case of no claim of identity, the feature set is compared against all the biometric reference templates on file. In case of an implicit negative claim of identity, the feature set is compared with a smaller set of biometric reference templates that are associated with the claimed identity. The identity of the individual is established if the feature set matches any of the stored reference templates in either of the two cases.

Verification is a one-to-one process. In verification, an individual submits biometric sample(s) and makes a positive claim to an enrolled identity (AHGBEA, 2007, p. 33). The sample(s) are processed through the signal processing system to obtain a feature set .The feature set is compared against the biometric reference template on file for the claimed identity. If the feature set matches the reference template, the identity of the individual is established.

2.1.2. Biometric Architectures

The architecture of any system depends on the layout of its different components. The two basic architectural decisions in biometric systems are the locations of the biometric matcher and the template storage (AHGBEA, 2007, p. 46). Table 2.1 presents the possible locations for these subsystems. The server is a centralized or distributed system that is at a remote location from the requesting client. The local workstation is a computer platform from where a user initializes his recognition process. The peripheral device is a sensor unit that is connected to the local workstation through an interface. Physical token refers to smartcards, USB sticks or other such objects.

Table 2.1 Biometric Matching and Storage Locations (AHGBEA, 2007, p. 46)

Storage Location	Matching Location
Server (Central/Distributed)	Server
Local Workstation (Client)	Local Workstation (Client)
Device (Peripheral)	Device (Peripheral)
Physical Token	Physical Token

From the information provided in Table 2.1, 4*4 = 16 storage and matching configurations are possible. These are shown in Table 2.2. Factoring on prescribed assurance levels, and commonly observed biometric solutions, six out of these 16 configurations were recommended for use in live implementations (AHGBEA, 2007, p. 50). These are marked with an X in Table. 2.2.

Table 2.2 Matrix of Biometric Storage and Matching Configurations (AHGBEA, 2007, p. 50)

Matching/Storage Configurations	Storage on Server	Storage on Client	Storage on Device	Storage on Token
Matching on Server	X			X
Matching on Client		X		
Matching on Device			X	X
Matching on Token				X

2.2. Fingerprint Recognition Systems

Among the different biometric systems, fingerprint recognition systems are the most widely deployed. They are quite popular due to the associated ease in acquisition, availability of multiple sources for collection and their established use with law enforcement (National Science and Technology Council Subcommittee on Biometrics and Identity Management, 2006). According to IBG (2009), fingerprint recognition systems accounted for close to 67% of the annual revenue generated by the entire biometrics market for the year 2008, where 29% was from commercial fingerprint recognition systems and the rest from Automated Fingerprint Recognition Systems (AFIS).

2.2.1. Fingerprints

A fingerprint is a representation of the layer of skin or epidermis of a finger. It is made of interleaved ridges and valleys (Ashbaugh, 1991; Maltoni, Jain, & Prabhakar, 2009, p. 97). The formation of the fingerprint is through a combination of genetic and environmental factors in the various stages of fetus development (Maltoni & Cappelli, 2008, p. 23). While the genetic code is instructed by the body for a general mode of formation, random conditions induced by the environment introduces specificity, as a result even identical twins have different fingerprints (Jain, Prabhakar, & Pankanti, 2002). The pattern becomes stable at seven months, the configuration being permanent from there on with the exception of accidents like cuts to the fingertips (Babler, 1991; Maltoni et al., 2009, p. 97). These two properties, uniqueness and permanence make fingerprint an excellent candidate for use as a biometric identifier.

2.2.2. History of Fingerprint Recognition

2.2.2.1. Fingerprints in Ancient Times

The earliest evidence for cognizance of humans to patterns on their fingertips dates as far back as 7000 B.C. Bricks found from the Neolithic Age in the ancient city of Jericho had their "surfaces impressed with a herringbone patterns by a pair of prints of the bricklayers thumbs, giving a keying as provided by hollow in modern bricks" (Berry & Stoney, 2001, p. 8). The first proper reference to the use of fingerprints can be traced to ancient Babylon in 2000 B.C. Clay pottery and cuneiform tablets were marked with the fingerprint impression of the person who made them to serve as the equivalent of a brand label (Ashbaugh, 1991, p. 19). The discovery of these cuneiform tablets in ancient Egypt may indicate that fingerprinting spread to other countries from Babylon (Ashbaugh, 1991, p. 19). China has a well documented history of the use of fingerprints since 800 B.C. They were included as part of official seals and legal proceedings, with the parties to the contract impressing their prints on the sheets where the contract had been written (Laufer, 2000/1912).

2.2.2.2. Early Scientific Research Into Fingerprints

The scientific research into fingerprints did not start till the 17th century. The modern foray began with the study of finger anatomy. In 1684, Dr. Nehemiah Grew, in a paper published in *Philosophical Transactions*, described sweat pores, epidermal ridges and their arrangements. His paper had a drawing of the hand that illustrated the ridge flow on fingers (Ashbaugh, 1991, p. 23). Govard Bidloo in 1685 published a book on human anatomy that showed the structure of friction ridges on the underside of the fingers and had a drawing that provided a reference to the arrangement of ridges (Ashbaugh, 1991, p. 23; Berry & Stoney, 2001, p. 16). Marcello Malpighi in 1686 explained the function of friction ridges for grasping objects and their morphology (Ashbaugh, 1991, p. 23; Berry & Stoney, 2001, p. 16). Close to a century later, J.C. Mayer in his comments on friction ridges and individuality from 1788 claimed that the friction ridge patterns could not be duplicated between two individuals (Ashbaugh, 1991, p. 23). This was one of the first statements on the uniqueness of fingerprints. He also explained the repetitiveness and similarity of friction ridge patterns, which are the two foundational principles of fingerprint identification (Ashbaugh, 1991, p. 23). In 1823, Joannes Purkinje defined nine different types of fingerprint groups based on his observation of ridge patterns (Ashbaugh, 1991, p. 23; Berry & Stoney, 2001, p. 19). This was one of the earliest attempts at classification of fingerprints.

2.2.2.3. Fingerprints for Identification: Herschel and Faulds Debate

The practical applications of fingerprint identification as initially documented were by William Herschel and Henry Faulds in the latter half of the nineteenth century. Herschel was the first to confirm ridge persistency, which is the reason for permanence of fingerprints (Berry & Stoney, 2001, p. 25). He took his own palm impressions in 1860 and in 1890 and saw that while the passage of time had allowed creases to run across his fingers, the sequence of ridge detail remained exactly the same (Berry & Stoney, 2001, p. 25). In a letter published in *Nature* in 1880, he explained how he had been collecting fingerprints for years and had convinced law enforcement officials in India to use fingerprints to identify criminal's twenty years earlier, in response to a competing claim by Henry Faulds (Berry & Stoney, 2001, p. 25; Laufer, 2000/1912, para. 4).

Henry Faulds was a British doctor working out of India and Japan. Inspired by finger impressions found on ancient shards of pottery, he became interested in fingerprints. In experiments conducted with the help of patients at a Japanese hospital where he worked, he showed that the re-growth of fingerprints was variable, but ridge patterns remained unchanged in the manner similar to that described by Herschel (Berry & Stoney, 2001, p. 29). He claimed that he had been helping Japanese officials to find criminals based on fingerprint identity for some time (Ashbaugh, 1991, p. 23; Berry & Stoney, 2001, p. 29). This included an incident at an embassy where he used a greasy finger mark to solve a petty theft (Cole, 2004, p. 2).

2.2.2.4. Development of Fingerprint Classification Systems: Faulds, Galton and Henry

Faulds made the earliest attempts at a fingerprint classification system. His system worked on the basis of syllables, where consonants represented a general pattern type while vowels represented the center of the fingerprint (Cole, 2004, p. 3). Each fingerprint was then a word, and these words could be arranged in an indexed alphabetical order. Unfortunately, Faulds attempts to get the Scotland Yard interested in his system met with "little success" (Cole, 2004, p. 3).

Sir Francis Galton's interest into fingerprints started when Faulds sent him a copy of the 1880 letter where he made a claim on fingerprint identification (Berry & Stoney, 2001, p. 31). Galton was at that time an expert at the Bertillon system, an identification mechanism based on anthropometric measurements (Berry & Stoney, 2001, p. 32). Galton started working with a prior classification done by Purkinje, and found it ineffective. He devised his own refined groupings, and ended up with 60 types (Cole, 2004, p. 4). He realized the difficulty in working with some many classes and reduced all types into three common classes: arch, loop and whorl (Berry & Stoney, 2001, p. 33; Cole, 2004, p. 4). He also introduced the notion of ridge features or minutiae for use in fingerprint classification. While his classification did not improve on the existing search capability with the Bertillon system, it was incorporated for use in Bertillon cards (Morland, 1950).

Sir Edward Henry improved on the work done by Galton and with the help of his assistants developed the Henry classification system for ten-prints. He started with the basic classes as Galton, but added other groups for rare patterns (Cole, 2004, p. 5). The system relied on the principle of serial categorization and used three levels of rules called primary, secondary and sub-secondary classifications (Cole, 2004, p. 5). In primary classification, each finger was characterized based on whorls while in secondary classification arches and loops where used. The sub-secondary classification used ridge counting for loops and ridge tracing for whorls (Cole, 2004, p. 6). Henry, who was a colonial officer in India, introduced the system in his jurisdiction in 1895, and it was adopted throughout India as a replacement to the Bertillon system with its primary use in awarding pensions (Berry & Stoney, 2001, p. 26). Around the same time the Henry system was developed, a team led by Dr. Juan Vucetich in Argentina developed another system based on Galton’s classification. The Vucetich system had a significant overlap with the classifications used in the Henry system, with the main difference being the division of sub pattern types for whorls (Cole, 2004, p. 5). Almost all subsequent fingerprint systems were based on either the Henry system or the Vucetich system or a combination of them (Cole, 2004, p. 15).

2.2.2.5. Related Developments

Fingerprint systems did not immediately replace existing system based on anthropometric measurements that were used for criminal identification. It initially gained adoption in civilian areas as a cheap replacement for institutions and bureaus that could not afford an anthropometric system (Cole, 2004, p. 14). Soon problems with the existing anthropometric system became demanding, specifically the need for skilled people to take exact physical measurements. Fingerprint systems shifted the requirements for skilled people to the back of the system and required a simple rolled impression for acquiring prints that could be done by less-skilled people (Cole, 2004, p. 14). Over time, the dominance of anthropometric systems was gone and fingerprint systems became the standard for criminal identification. Between 1900 and 1930, fingerprint systems were adopted worldwide, each slightly different from the other, under different names such as the Gasti system in Italy and the Daae system in Norway (Cole, 2004, p. 15). As early as the 1919, with the growing success of manual fingerprint classification system and increasing sizes of fingerprint databases, the need for automation in fingerprint classification was recognized (Cole, 2004, p. 15). The earliest application of automation to fingerprint identification was the use of IBM punch card sorters that encoded an individual's classification information and could be indexed to allow for a card search. However this just solved part of the problem (Cole, 2004, p. 18). The U.S. Federal Bureau of Investigation (FBI) designed and developed of AFIS in 1972 (Berry & Stoney, 2001, p. 36; Cole, 2004, p. 18). The AFIS system resolved many of the issues associated with manual identification processes like acquisition, feature extraction and pattern matching.

2.2.3. Fingerprint Acquisition

The acquisition module is the first point of interaction between an individual and the fingerprint system. It has an important influence on the performance of fingerprint recognition. Any inconsistencies or errors that are introduced here get propagated throughout the system.

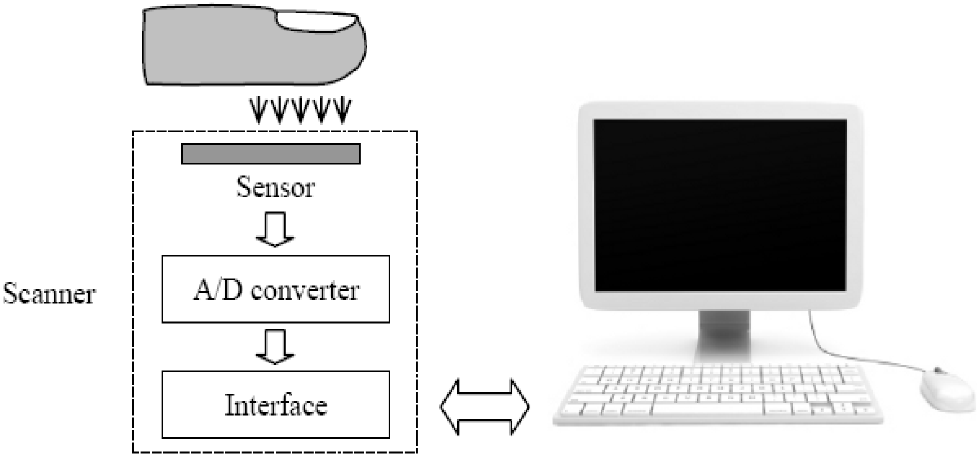

Figure 2.2 Block Diagram of a Fingerprint Scanner (Maltoni et al., 2009, p. 58)

Between the first use of fingerprints for law enforcement and the development of AFIS, the ink technique was the method of choice for fingerprint acquisition (American Dermatoglyphics Association, 1990). An individual's fingers were spread with black ink and rolled on a paper to collect an impression. The impression was then scanned into a digital image. This kind of acquisition process was called offline acquisition. It changed with advances in storage mechanisms and the advent of automated recognition. In the present day, acquisition is done through a live fingerprint scan where the digital fingerprint image is acquired directly from the fingerprint surface (Maltoni et al., 2009, p. 57). The basic structure of a fingerprint scanner is shown in Fig 2.2.

The sensor is the most important part in a fingerprint scanner since it reads the fingerprint ridge pattern as an analog signal. There are three types of sensors that are commonly used: optical, electrical and thermal. Optical sensors are the most popular among all the fingerprint sensors. Different acquisition techniques are used in optical sensors. Maltoni et al. (2009, pp. 58-59) list five such methods. These are briefly described here.

1. Frustrated Total Internal Reflection (FTIR): This is the oldest and most commonly used process. The finger touches one side of a glass prism, while the other side is illuminated using a diffuse light. The light

entering the prism is reflected by the valleys and absorbed by the ridges, allowing for discrimination between them (O' Gorman & Xia, 2003). A Charged Coupled Device (CCD) or Complementary Metal Oxide Semiconductor (CMOS) image sensor then captures the reflected light. The working of an FTIR based optical sensor is shown in Fig 2.3.

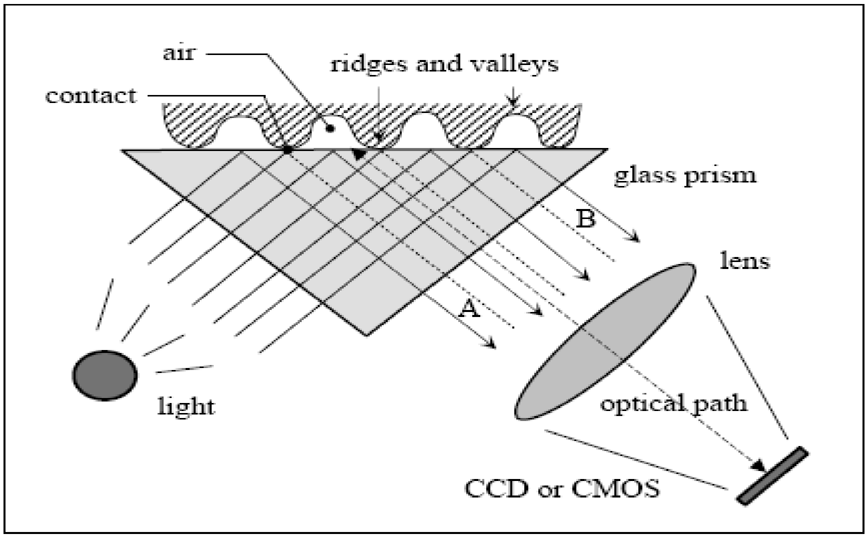

Figure 2.3 FTIR Based Optical Fingerprint Sensor (Maltoni et al., 2009, p. 63)

2. FTIR with a sheet prism: Instead of using one large prism for FTIR, the size of the design is reduced by using small prismlets adjacent to each other to form a sheet layer which is then illuminated with diffuse light (O' Gorman & Xia, 2003).
3. Optical fibers: The prism and lenses are substituted with fiber-optical platen (Fujieda, Ono, & Sugama, 1995). The finger is in direct contact with the platen, and the CCD or CMOS image sensor is connected directly with the platen, receiving the fingers residual light as obtained through the glass fibers (Maltoni et al., 2009, p. 64).
4. Electro-optical: There are two layers. The first layer is a polymer that emits light when it is polarized with a voltage. The potential difference between ridge and valleys gives a representation for the fingerprints.

The second layer is attached to the first and is a layer of photodiode array that converts the light emitted by the first layer into a digital image (Young et al., 1997).

5. Direct Reading: a highly quality camera is used to directly focus on the fingertip, without the finger touching any surfaces (Parziale, 2007).

Capacitive sensor is a solid-state sensor and consists of an array of plates each of which is a tiny sensor by itself. The capacitive difference generated by placing the finger on the platen is measured and converted to pixel values to form an image. Thermal sensor is another form of solid-state sensor. In the case of thermal sensors, the heat flux is measured and converted into a digital representation of the fingerprint surface.

2.2.4. Fingerprint Features

The digital image acquired through a fingerprint scanner represents a sample of the biometric trait. It is the starting point for the creation of templates, matching and all other processing within the biometric system.

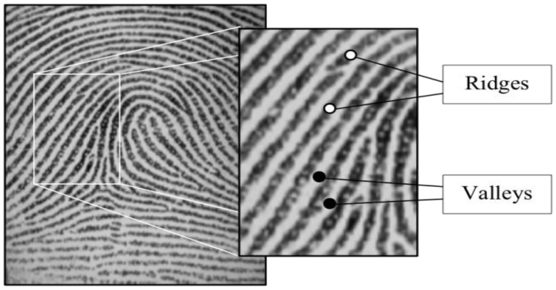

Figure 2.4 Ridges and Valleys (Maltoni et al., 2009, p. 97)

The fingerprint sample contains structural characteristics in the form of a pattern of interleaved ridge and valleys. Fig 2.4 shows these fingerprint characteristics. These ridge patterns represent details that are considered features of the fingerprint sample. The ridge details are present in a hierarchy of three levels, corresponding to the three types of fingerprint features as discussed by Maltoni et al. (2009, p. 97).

1. Global level features: ridge and valleys typically run smooth, but in some regions of the fingerprint they form distinctive shapes characterized by high curvature and frequent ridge terminations. The observed shapes are called singularities. The shapes are of three types: whorl, loop and delta. These are shown in Fig. 2.5.
   A landmark defined by the north most point of the innermost ridgeline is used for pre-alignment of images during fingerprint matching (Maltoni et al., 2009, p. 98). This is a called a core point, and is shown in Fig. 2.5. Global level features are typically used for fingerprint classification in databases in order to make search and retrieval easier.

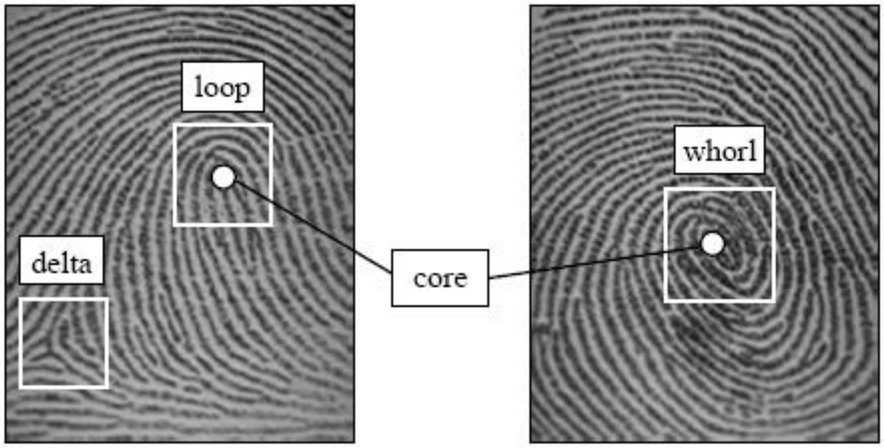

Figure 2.5 Loop, Delta, Whorl and Core (Maltoni et al., 2009, p. 98)

2. Local level features: When seen in small detail, the ridgelines become discontinuous in certain parts of the fingerprints. The different ways in which this can happen is called minutiae. There are seven possible types that exist for minutiae (Maltoni et al., 2009, p. 98). These are shown in Fig.2.6.

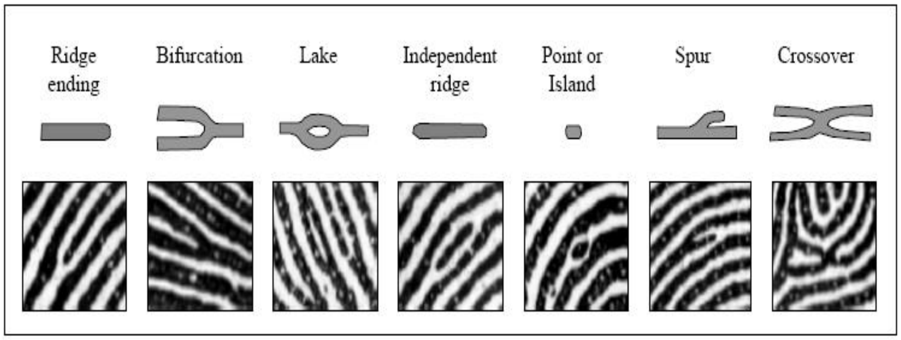

Figure 2.6 Types of Minutiae (Maltoni et al., 2009, p. 99)

The two types most commonly observed in fingerprints are ridge endings (ridge suddenly comes to an end) and ridge bifurcations (the ridge divides into two). Minutiae features are quite useful for matching in fingerprint recognition systems since the correspondence of a small number of minutiae is enough to say with a high confidence that two fingerprint impressions are similar.

3. Very local features: these include ridge attributes such as width, size, shape, pores etc. While they provide a high level of distinction for matching, current scanners are not powerful enough to detect them (Maltoni et al., 2009, p. 101).

2.2.5. Fingerprint Feature Extraction

The purpose behind feature extraction is to either provide a feature set for use in creating templates, matching or as an intermediate step for other processing.

A feature extraction process for fingerprints can be typically divided into a number of different steps. These steps vary from vendor to vendor.

1. Segmentation: the fingerprint area (foreground) is separated from the background in the fingerprint sample. This is useful since it prevents extraction from noisy background regions, allowing for reliable processing of features (Maltoni et al., 2009, p. 116). Previous work on segmentation includes the use of fingerprint block gradients by Mehtre,

Murthy, Kapoor and Chatterjee (1987), use of Gabor filters in combination with a clustering algorithm by Jain, Ratha and Lakshmanan (1997) and the use of pixel coherence, mean and variance by Bazen and Garez (2001).

2. Local ridge orientation: The local ridge orientation for a pixel is the angle that the fingerprint ridges crossing through an arbitrarily small neighborhood centered on the pixel form with the horizontal axis (Maltoni et al., 2009, p. 102). Finding local ridge orientations for all pixels is computationally intensive; so it is typically calculated for distinct positions. A direction or orientation map encodes the local orientations for the ridgelines as a matrix. It is quite useful for identifying singularities and minutiae. Previous work on finding local orientations and orientation maps include use of sinusoidal modeling with variation theorem by Maio and Maltoni (1998) and the use of principal component analysis by Bazen and Garez (2000).
3. Singularity detection: the core and delta points are detected from the fingerprint by using the orientation map. Previous work in finding singularities involves calculation of the Poincare index with orientation map smoothing by Karu and Jain (1996) and the use of coherence operators by Cappelli, Lumini, Maio and Maltoni (1999).
4. Enhancement: This step improves the quality of the fingerprint. It makes the ridge structures more clear in the fingerprint regions that can be recovered and marks the rest as noise. Previous work on enhancement includes calculating the normalization of intensity value for each pixel by Hang, Wan and Jain (1998).
5. Minutiae detection: This step identifies the discontinuities in the ridgeline flow. This is either performed directly on the sample or after the sample has been binarized and the ridgelines within the binarized sample are thinned. A post-processing step may also be involved to remove spurious minutiae. A couple of experiments related to minutiae detection are given in 2.2.5.1.

2.2.5.1. Experiments Related to Minutiae Detection

Maio and Maltoni (1997) proposed a method for minutiae detection that worked directly on the gray scale image. The concept was to track the ridge pattern by using its local orientation. Given a starting point and angle, an iterative algorithm was devised in which a new section of a ridgeline was identified and the local maxima of this section orthogonal to the ridge direction were calculated. By connecting the consecutive maxima for the ridge sections, a polynomial approximation of the ridgeline was obtained. Traversing along the ridgeline, if it ended or bifurcated at some point, the algorithm stopped and returned the minutiae characteristics. Using this algorithm on all the ridgelines in the ridge pattern, all the minutiae were obtained. In order to prevent multiple tracking of ridgeline flow, and detect false minutiae, an auxiliary sample was used to keep track of ridgeline intersections. An experiment was performed on fingerprint samples from the National Institute of Science and Technology (NIST) fingerprint database and FBI sample set, where their method was compared with four popular binarization based methods and it was concluded that their method was faster and worked better at detecting false minutiae.

MINDTCT is a minutiae detection algorithm that was designed by NIST. It uses a binarization algorithm before performing minutiae detection. In binarization, for each pixel in the sample, the ridge flow is detected in its block by comparing the sample and pixel intensities in the neighborhood of the block, and a binary value is assigned to the pixel based on whether a ridge flow is detected or not. The binarized sample is then used to identify local pixel patterns. These are then compared to a candidate list of patterns and associated candidate minutiae. If there is a match, the candidate minutiae are detected through correspondence with this association. From the list of candidate minutiae, the false minutiae are pruned and any other unwanted observed features are removed.

2.2.6. Fingerprint Templates

During enrollment, fingerprint sample(s) are acquired and processed to obtain a set(s) of fingerprint features. When a set(s) of fingerprint features attached to an identity is stored on a data medium within the biometric system, it is called fingerprint template(s).

2.2.6.1. Composition of Fingerprint Templates

The different features that can compose a fingerprint were previously described in section 2.2.4. Every feature has numbers or labels associated with it, which is its representation (ISO, 2008, p. 4). When the features used in the fingerprint system are minutiae points, there are three types of information according to Hill (2001) that are seen in the representation:

1. Location: position of the minutiae as x-coordinates and y-coordinates within the fingerprint sample calculated relative to the origin of the system.
2. Orientation: angle of the vector made by a ridgeline passing through the neighborhood of minutiae.
3. Type: The manner in which the ridgeline becomes discontinuous to give minutiae. This was previously discussed in section 2.3.4.

These three constitute the basic form of the representation for minutiae features. There is an extended representation for minutiae where optional types of information such as quality, ridge counts, ridge curvatures and singularity locations are also included. All the different types of information are limited by the set of values that they can take. These range definitions are provided in minutiae standards. Three popular standards for minutiae based fingerprint templates are ANSI/INCITS 378 that is described in InterNational Committee for Information Technology Standards (2004), ISO/IEC 19794-2 that is described in ISO (2005) and ANSI/NIST-ITL 1 that is described in NIST (2007). The following provides the range that each type of minutiae information can take as given in the three standards:

1. ANSI/INCITS 378: It has termination or bifurcations for type. It expresses location in pixels that are derived from a Cartesian system. It records quality in the range from 0 to 100, with 0 for minimum and 100 for maximum quality. It has orientation in units of 2 degrees.
2. ISO/IEC 19794-2: It has termination, bifurcation and other for type. It records quality scores in the range 1 to 100, with 1 for minimum and 100 for maximum quality. It has orientations in units of 1.40 degrees. It expresses location in pixels with the origin of the coordinate system at the upper left hand corner.
3. ANSI/NIST-ITL 1: It has four values for type: termination, bifurcation, compound (trifurcation or crossover) and undetermined. It permits quality values in the range 0 to 63: 0 indicates that the minutiae is manually encoded, 1 that there is no measure for quality, and values from 2 to 63 represent the quality scores with decreasing level of confidence, with 2 showing the highest confidence. It uses millimeters for expressing location rather than pixels. The origin of the coordinate system used is at the lower left hand corner.

The fingerprint template is then a record with multiple fields, where each field contains the representation of single minutiae in the fingerprint. The total number of fields is the number of features in the feature set that constitutes the fingerprint template. The size of the template is based on the size of the fields. The fingerprint template may also have other data attached to it for formatting in order to use for exchange between fingerprint systems. The following describes the ISO/IEC 19794-2 and the ANSI/NIST-ITL 1 minutiae based fingerprint templates as given in ISO (2005) and NIST (2007):

1. ISO/IEC 19794-2 template: has a generic format designed for automated fingerprint recognition. It can include data from single or multiple fingerprint impressions. The format is called a minutiae record. The minutiae record has a record header with general information like image size and resolution and the number of fingerprint impressions represented also called finger views. For each finger view there is a corresponding single finger record that is further divided into subfields,

each subfield containing information on a single minutiae in either basic or extended representation.

2. ANSI/NIST-ITL 1 template: has a format designed for latent search of prints as well as applications of physical or logical access control. The format is called a type 9 minutiae data record. It can include data from single or multiple impressions of a finger. In its standard form, the record has twelve fields each of which is recorded as text. The initial 11 fields contain general information like record length, impression type, minutiae format, originating system, singularities etc. The last field has data on minutiae and ridge count. The last field is divided into many subfields, each of which is dedicated to a single minutiae and contains representation in either basic or extended form. A type 9 record can be stored as part of a meta-record file, where each file contains multiple records, each for a different finger.

2.2.6.2. Creation of Fingerprint Templates

There are no standardized processes that are followed in the creation of a fingerprint template. Every vendor has their own method of template creation that depends on the proprietary modules that they use. However, a general outline of these methods can be derived as done below:

1. Fingerprint selection: Most fingerprint systems take multiple fingerprint samples from the user during enrollment, to ensure invariance in the fingerprint data (Uludag, Ross, & Jain, 2004). They may also create more than one fingerprint template for the same purpose. The mapping definition between the fingerprint sample(s) and the fingerprint template(s) is an important focus point within the fingerprint system and is referred to as fingerprint selection (Uludag et al., 2004). This mapping can be defined in different ways (Li, Yin, Zhu, Hu, & Chen, 2008). In one mapping definition, all the samples are combined to form a super-sample and features are extracted from it to get a single template. In another, features are extracted individually from each

sample and then combined to get a single template. In a third definition, the features are individually extracted from each sample and a corresponding template created for each.

2. Fingerprint feature extraction: Once the mapping definition between the sample(s) and template(s) is fixed, the extraction process creates fingerprint feature set(s) based on this mapping. A description of this feature extraction procedure was given in section 2.2.5.
3. Fingerprint template processing: Once the feature set(s) are obtained from the feature extractor, they need to be structured into a record. The structuration process is not fixed and varies depending on what type of feature is being stored. The structured feature set is the fingerprint template.
4. Fingerprint template storage: The fingerprint template is then transmitted to be stored in one of the many storage locations described in section 2.1.2. This can be done in the open or using cryptographic protocols such as RSA or DSA.
5. Fingerprint template update: This is an additional step that is incorporated either as part of the extraction process or during verification. Fingerprint traits can vary due to age or environmental factors. To account for change in biometric traits and ensure stability in matching performance, existing template(s) need to be modified to include information from a more recent instance of the fingerprint sample(s) (Uludag et al., 2004).

2.2.7. Fingerprint Matching

In fingerprint matching, the representations of two fingerprints are taken and compared to either return a score as a degree of similarity indicating how well they fit together or a binary decision of match or non match. The representation of the fingerprint can be the sample itself or the feature set / template extracted using the feature extraction process described in section 2.2.5.

Fingerprint matching approaches belong to three categories according to Maltoni et al. (2009, p. 171):

1. Correlation based matching: this is used when the representation is the fingerprint sample. The two fingerprint samples are superimposed on each other and a pixel wise calculation of their correlation value is performed as their corresponding alignment is varied.
2. Minutiae based matching: this is used when the representation is a template or a feature set in terms of minutiae that are extracted from the fingerprint samples. The basic idea is defined in the minutiae-matching problem, which finds the alignment between representations such that the number of minutiae pairings is maximized. An alternate notion is the visualization of the minutiae matching as point pattern matching problem. Minutiae-based matching is the most commonly used among the three categories presented here. A couple of experiments related to minutiae based matching is given in section 2.2.7.1.
3. Non-minutiae feature based matching: as with correlation based matching, this is typically used when the representation is the fingerprint sample. It Identifies ridge information such as texture, orientation etc in the fingerprint representations and uses them in complex calculations for matching, the type of calculations varying depending on the type of information used. This type of matching is especially useful when working with fingerprint samples of low quality from which reliable minutiae extraction is difficult.

2.2.7.1. Experiments Related to Minutiae based Matching

Ratha, Karu, Chen and Jain (1996) described a minutiae based matching algorithm that used the Hough Transform. There algorithm had three steps: registration, pairing and score computation. In registration, it was estimated that the two fingerprint representations were same, and that one could be obtained from the other using a similarity function consisting of transformation parameters of rotation, scale and translation. All the possible transformation was identified. In pairing, each transformation was constructed as a Hough Transform and then applied to each minutia pair, to get a similarity score within a margin of error. In score computation, the matching scores were collected from all pairs for a transformation in an accumulator array, and the transformation, which maximized the total number of matched pair, was declared as the best one. The algorithm finally returned a list of 10 fingerprints as candidate matches. An experiment was performed using samples from the NIST-9 database, with an accuracy of 80% at 10 % False Reject Rate and the conclusion that their matching was being done at fast speeds.

A fingerprint matcher called BOZORTH3 is included in the NIST Biometric Image Software distribution. It uses a modified version of an algorithm developed by Allan S. Bozorth while at FBI. The input to the matcher is a set of files that contains the x-coordinate, y-coordinate, and the orientation of the minutiae point to match the fingerprints. There are three main steps to the algorithm. First, the minutiae features are extracted from both fingerprint images and an intra-fingerprint comparison table is created for each. To construct the table, pairs of minutiae within the fingerprint that are sufficiently close to each other are compared. Second, comparing the intra-fingerprint tables for the two fingerprints generates an inter-fingerprint table. To construct this table, the pair-pair distance and pair-pair angle fields are compared to see whether they are compatible within a predefined threshold. Finally, the inter-fingerprint table is traversed and the table entries are linked to form a forest of clusters. The similarity score is the number in the minutiae in the largest cluster.

2.3. Attacks on Biometric Systems

A biometric system is susceptible to harm by external users or malicious insiders. This is possible due to the existence of vulnerable points in different parts of the system. Ratha, Connell and Bolle (2001a) identified eight locations that can be attacked in a biometric system. These are given in Fig. 2.7.

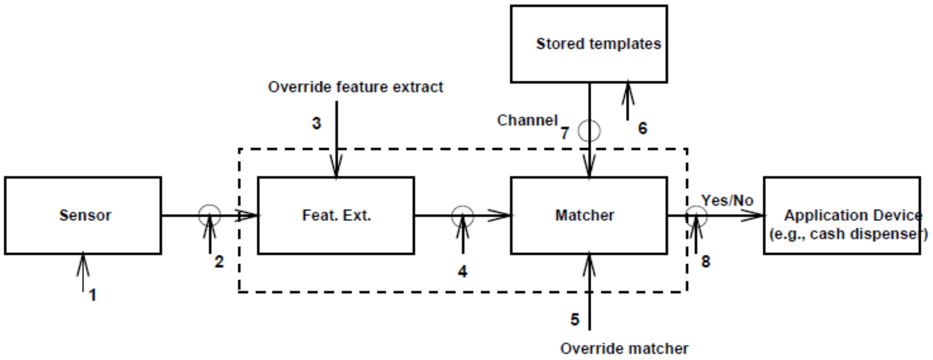

Figure 2.7 Possible Attack Points in a Generic Biometric System (Ratha et al., 2001a, p. 224)

2.3.1. Classes of Attacks

Based on the availability of these vulnerable points, the attacks on biometric systems can be classified in two broad categories: direct and indirect attacks.

2.3.1.1. Direct Attacks

A fake physical form of the biometric trait is reproduced with or without the help of the real user and submitted to the sensor. These are also referred to as Type 1 attacks (Ratha et al., 2001a, p. 224). Direct attacks include all intrusions launched at location 1 in Fig 2.7. The feasibility of these attacks is very high. There are two reasons. First, they do not require knowledge of the biometric authentication system or any of its access privileges. Second, they work in the analog domain close to the end user that makes it impossible to user common digital protection mechanisms like encryption or hashing.

2.3.1.2. Indirect Attacks

The knowledge of the functionalities within the biometric system is exploited to deceive one or more of its components. These attacks are possible only if the malicious user has access to the system components like matcher or feature extractor and/or privileges in this regard. According to Ratha et al. (2001a, p. 224) these attacks can be further divided into seven classes:

1. Type 2 attacks: Using an old captured biometric signature to perform a replay by submitting it to the authentication system, at the point of communication between the sensor and the feature extractor.
2. Type 3 attacks: Overriding of feature extractor by using a Trojan horse where the malicious user manipulates the feature values that get returned in the authentication process.
3. Type 4 attacks: Tampering with feature representation in the communication between the extractor and the matcher where genuine feature values are replaced with the one selected by the attacker.
4. Type 5 attacks: Overriding the matcher module by using a Trojan horse where the attacker manipulates the score calculation to produce an artificially high matching score as returned in the authentication process.
5. Type 6 attacks: Gaining access to the template database and obtaining privileges to add a new template and remove or modify an existing template.
6. Type 7 attacks: Monitoring the transmission medium between the template database and matcher and to alter the transmitted templates.
7. Type 8 attacks: Overriding the result that is returned by the decision module.

Indirect attacks include all intrusions launched at locations 2-8 in Fig 2.7. While they are less applicable than direct attacks, they can be conducted a lot faster since there is no physical reproduction involved in the process. Unfortunately, as these attacks are in the digital domain they can be thwarted using encryption, hashing and other such schemes.

2.3.2. Type 2 Attacks on Biometric Systems

In such attacks, a malicious user replays a previously used biometric signature to the authentication system. The replayed signature can be one of two objects:

1. Old real sample: the replayed signature is an old biometric sample that was sent from the sensor to the extractor either in encrypted or unencrypted form. An attacker who monitors the communication link between the extractor and the sensor can read the unencrypted/-encrypted data from the line. In the case where the biometric sample is not cryptographically protected, it is trivial for the attacker to obtain it. If the transmission were done using a weak encryption mechanism, it would be relatively easy for the attacker to infer the biometric sample.
2. Synthetic approximate sample: The replayed signature is artificially made by using match scores, using biometric templates, or a combination of both. When match scores are used, it is called the hill-climbing method, when templates are used it is called the masquerade method. In case of hill climbing, there is an iterative process, where an initial fingerprint sample is updated by using match scores obtained by passing the sample through the biometric system. In case of the masquerade method, the data given in the template is used to reconstruct information about the different attributes of the biometric sample, such as the ridge flow and singularities in case of fingerprints, in combination with general information available about the biometric modality.

The replay of biometric signature is done at the link between the sensor and the feature extractor. This communication medium between the sensor and the feature extractor is shown in Fig 2.7 as location 2. The feature extractor takes the replayed signature and extracts a feature set from it. This feature set is then compared with the existing biometric templates in the system by a matcher either in a verification or identification mode. If a match result is obtained, the decision module gives access to the attacker as if he were an authorized user of the biometric system.

2.3.3. Masquerade Attacks: Type 2 Attacks that Reconstruct Samples from Templates

Type 2 attacks where the approximate synthetic samples are created using the masquerade method are called masquerade attacks. The masquerade method is a technique for recreating samples using templates. The method is generic and can be applied to templates from all biometric modalities. It is a three or four step process, as shown in Fig 2.8. The process was described first by Hill (2001, pp. 36-40).

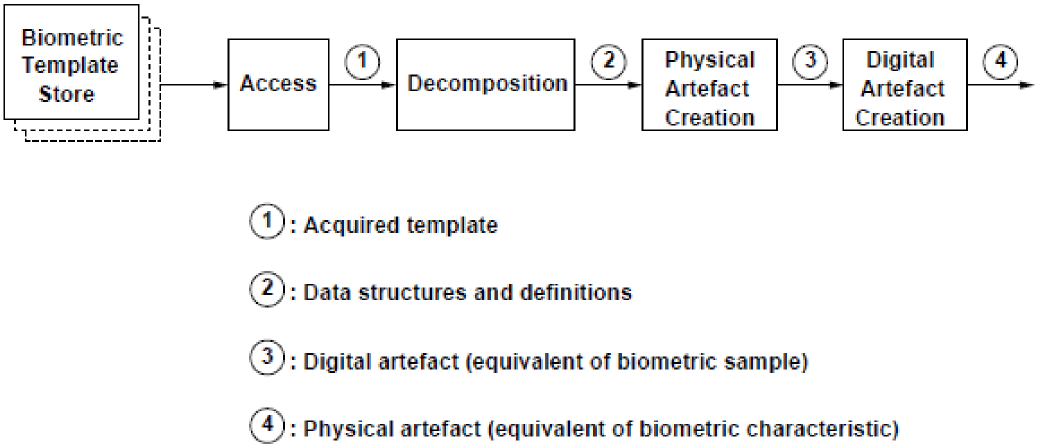
Biometric Template Store
Access
1
Decomposition
2
Physical Artefact Creation
3
Digital Artefact Creation
4
1 : Acquired template
2 : Data structures and definitions
3 : Digital artefact (equivalent of biometric sample)
4 : Physical artefact (equivalent of biometric characteristic)

Figure 2.8 Masquerade Method (Hill, 2001, p. 36)

Each of the steps in this method is briefly described in the following section.

1. Template access: In this step the malicious user finds a weakness in the management of the biometric templates and uses it to their advantage. This is possible in two ways. First, the malicious user can intercept the communication between the template storage and the matcher as the template is getting transmitted during enrollment. Second, the malicious user can try to compromise an external system

that implements the access control mechanism for the template storage and use that to gain access to the storage location. The difficulty of using either method depends on the security measures used in communication in the system as well as the type of storage location used. For example, it is easier to exploit a storage location that is a centralized database and is public than a location that is a self contained token that is kept in secret by an individual. The advantages and disadvantages of using different storage location for keeping the template safe are discussed in Hill (2001, pp. 25-31) and AHGBEA (2007, pp. 47-49).

2. Template decomposition: The biometric template is structured before being stored. The formatting is not consistent, and involves any combination of encryption, compression or processing (Hill, 2001, pp. 32-33). Encryption and compression are typically done using standard cryptographic techniques such as RSA and SHA; while processing is performed using private structures defined in standards such as ANSI/INCITS 378 for minutiae based fingerprint templates. In each of these cases, the formatting is required to be decomposed to reveal the template data actually stored, which is the second step of the masquerade method. The difficulty associated with the decomposition depends on whether multiple templates are available to infer the formatting structure or whether the formatting system/standard and its details are publically available or not (Hill, 2001, p.32).
3. Digital artefact creation: in the third step, the decomposed template data is used to recreate a likeliness of the original sample from which the template was derived. This is relatively less difficult in biometric modalities where the artefact is an image, such as fingerprint. Some generic information on the shape and formation characteristics of the biometric modality is expected in the recreation process (Hill, 2001, p. 38). It is this digital artefact that represents the replayed signature as used in a Type 2 attack as described at the start of section 2.3.2.

Some prior and current research done on reconstruction from minutiae based fingerprint templates is given in section 2.3.3.1.

4. Physical artefact creation: This is an additional step that is sometimes utilized in the masquerade method when creation of the digital artefact is not sufficient. For a discussion on creating physical artefacts from digital artefacts, refer to Galbally, Cappelli, Lumini, Maltoni and Fierrez-Aguilar (2008) and Galbally et al. (2010).

2.3.3.1. Experiments Related to Sample Reconstruction from Minutiae based Templates

Hill (2001) was the first to develop a reconstruction scheme for minutiae based fingerprint templates. He used a demonstration product made available with a commercial fingerprint matcher for this purpose. The formatted templates were acquired from a local hard drive and decomposed through the addition or removal of single minutiae points and modification of their orientation values to get minutiae’s, core and deltas. The shape of the fingerprints was predicted using a set of 23 heuristic relations between the minutiae points and this was combined in a fully connected neural network. Dividing the image into blocks and calculating complex numbers for each block using the core and delta positions determined the orientation map of the fingerprints. This orientation map was then used with the neural network to generate a sequence of splines passing through the minutiae points, essentially creating synthetic fingerprint images. A small database of 242 fingerprint samples was used in the experiment, 142 of which were from the Fingerprint Verification Competition (FVC) 2000 package and the rest created using a random fingerprint generator. A classification accuracy of 71% was observed.

Ross, Shah and Jain (2005) used fingerprint template consisting of only minutiae information to reconstruct the fingerprint. The templates did not contain singularity information as in the case of Hill (2001). The orientation map was created in three steps: generation of minutiae triplets, orientation estimation as weighted sum of orientations of a pixel in the region defined by the minutiae triplets, and finally averaging using a local filter. The minutiae data was then used to estimate class of the fingerprint through detection of the registration point and using only the minutiae around it for identification of eleven local and global features. Gabor like filters was then used in congruence with the orientation data and the ridge information relative to the fingerprint class to create the fingerprint images. Ross, Shah and Jain (2007) extended the previous work and used streamlines and line integer convolution for generating the ridge structure instead of Gabor filters. The streamline was constructed through identification of a seed point and using it in congruence with the orientation field. Line integer convolution was used for providing texture-based appearance to the ridgelines from the streamline construction. In both cases, the experiment was conducted on the NIST-4 database, with low identification rates between 30 and 40 percent.

Cappelli, Lumini, Maio and Maltoni (2006) developed an approach to reconstruct fingerprint images from ISO/IEC 19794-2 fingerprint templates defined in ISO (2005). It used the template information to estimate the fingerprint area, orientation image and ridge pattern. The fingerprint area was estimated by calculating the minimum area required to enclose all the minutiae in the template within an elliptical model. The orientation image was estimated by using the local direction for each minutia to find the parameters of an orientation model. The ridge pattern was created from the orientation image, the minutiae and a constant frequency value by using Gabor filters. The approach was used in an experiment with the FVC 2002 databases. Although a high similarity was observed between the original and reconstructed images, there was still spurious data produced during reconstruction. No performance results were provided. This approach was extended in Cappelli, Lumini, Maio and Maltoni (2007) to all fingerprint based standard templates.

Feng and Jain (2009) developed a novel fingerprint reconstruction algorithm, which worked by modeling the fingerprint image as a Frequency Modulation (FM) signal. The signal had a continuous and a spiral component, both of which were derived from the minutiae information given in the fingerprint template. Obtaining the dilated convex hull of the minutiae created the foreground mask for the fingerprint sample. The local ridge orientation was calculated by identifying the nearest minutiae’s. A phase offset was found for each fingerprint block using a nearest neighbor algorithm. The continuous phase of the fingerprint signal was constructed as piecewise planes at each block of the foreground mask through a combination of constant ridge frequency, the local ridge orientation and the phase offset. The spiral phase was a simple angular calculation on the positional value of the minutiae. Unlike the previous methods, this method created very few spurious minutiae and was able to recreate the entire fingerprint sample, rather than just a part of it. It also did not suffer from the minimum number of minutiae requirements seen in previous methods, and could be used for reconstruction even if only a single minutiae was available. An experiment was performed on the FVC 2002 fingerprint database, with an attack success rate of 70% for verification and 98.1% for identification.

2.3.4. Hill Climbing Attacks: Type 2 Attacks that Recreate Samples from Match Scores

Type 2 attacks where the synthetic approximate samples are created using the hill climbing method are called hill-climbing attacks. The hill climbing method is a technique developed for solving optimization problems. It can be used when there are a number of solutions, some of which are better than others. It involves starting with a sub-optimal solution to a problem (starting at the base of a hill) and then improving it in an iterative manner until a certain maximal condition has been reached (the top of the hill) where no more improvements can be made.

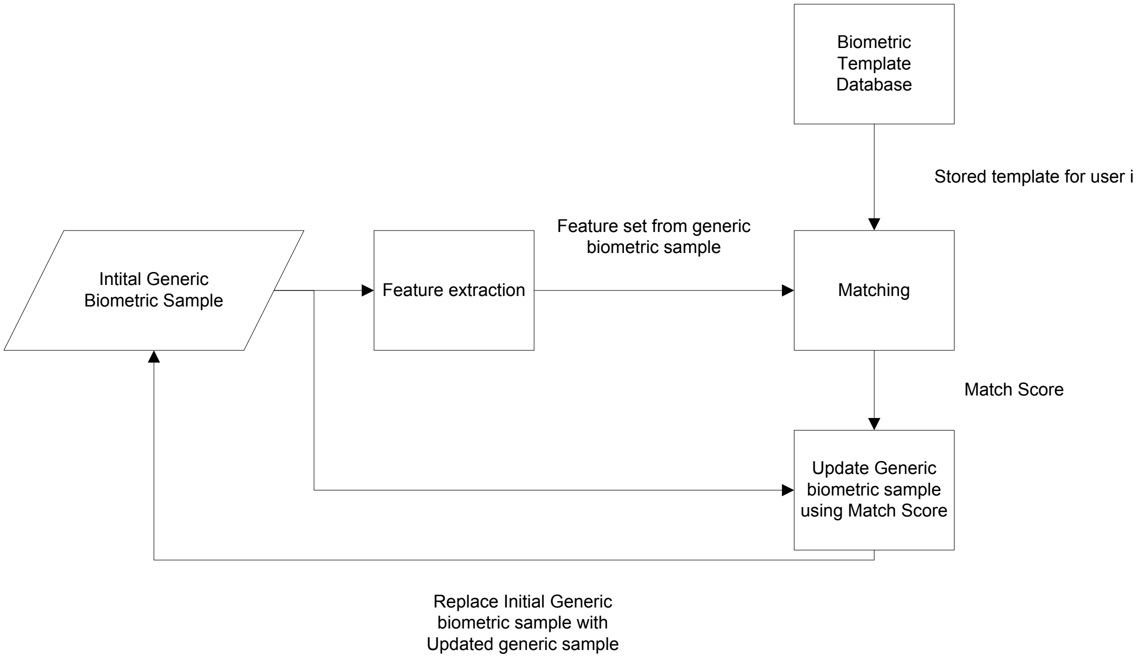
Biometric Template Database
Stored template for user i
Intital Generic Biometric Sample
Feature extraction
Feature set from generic biometric sample
Matching
Match Score
Update Generic biometric sample using Match Score
Replace Initial Generic biometric sample with Updated generic sample

Figure 2.9 Hill Climbing Method in Biometrics

The use of hill climbing for reconstructing a biometric sample was first suggested in Soutar, Gilroy and Stoianov (1999). The basic steps involved in the process are outlined in Fig. 2.9. There is an initial biometric sample that is created using random generation, inferred from previous database of biometric samples or obtained through the Masquerade method described in section 2.3.3. This initial sample undergoes feature extraction and is matched with an existing template of a user. A matching score is returned which is then used to update the features in the initial sample to obtain a modified sample. The modified sample then replaces the initial sample. These steps constitute a single iteration. Multiple iterations are run with the same steps. It is expected that each time the modified sample returns a better matching score than it did in a previous iteration. A point is reached when there is no more improvement in the matching score. The modified sample at this point is in its optimal state and represents the reconstructed sample.

2.3.4.1. Experiments Related to Recreating Samples from Match Scores

Soutar (2002) showed the practical application of the hill climbing method in a biometric recognition system as first suggested in Soutar et al. (1999). The application had two parts. In the first part, the recognition system was a simple design that worked with normal images and consisted of a matcher that was a phase only filter based correlator. Two images, one of an Apache helicopter and another of a space shuttle where taken, filters were created and these were matched with the images to get match scores. On the basis of these scores, a decision threshold was set. A filter of the space shuttle image was taken and matched with the Apache image in a simulation. In every iteration of the simulation, 64 randomly selected pixels were modified on their gray level in the Apache image and the output match score observed. Those set of pixel values which provided a positive score feedback were kept. At the end of 7 million iterations, the two images were similar to a certain scale. In the second part of the application, the two regular images were replaced with two fingerprint images from different users and matched as before.

Adler (2003) used the hill climbing method to reconstruct face images. Initially a generic face image was selected from a local database with the highest matching score. In iterations, the face image was modified using different eigenfaces (face templates) multiplied with constants to get a set of candidate images, which were cropped to ensure they were within the capacity for gray scale images. These were sent to the matcher to obtain corresponding match scores. The candidate image with the highest match score was the input for the next iteration. These iterations were repeated until no improvement in matching score was observed. The experiment was conducted with three commercial face recognition systems and 4000 iterations to get 99.9% matching scores on each. Adler (2004) later extended his previous work to include quantized matcher scores and encrypted face templates.

Ross et al. (2007) developed a framework for using the hill climbing method with the reconstruction process for fingerprint templates or the masquerade method. In this scheme, the reconstructed artefact was matched through the matching system with the minutiae template stored. The match score released by the process was then used as an input to the reconstruction process. No details were provided on how this incorporation would be done and consequentially no experimentation was performed in this regard.

2.4. Preventing Sample Reconstruction from Templates

The first and most important step in the masquerade method needed for sample reconstruction is template access. An effective way to prevent the reconstruction process is to protect the storage of biometric templates, thus ensuring that the templates do not get lost. There are two popular families of such template protection schemes, feature transformation and biometric cryptosystems. In both the template is used to create a secure reference, which is stored instead of the template itself, and used in processing for the biometric system.

2.4.1. Feature Transformation Schemes

Feature transformation schemes represent one family of biometric protection schemes. In feature transformation schemes, an invertible or non-invertible function is used to create a secure reference. The parameters of the transformation function are normally derived from a random key or a password. During authentication, the new raw data is taken to the transformation domain and matched with the secure reference. Feature transformation schemes are also referred to as cancelable biometrics (Ratha, Connell, & Bolle, 2001b). Fig 2.10 shows a basic design of such a system.

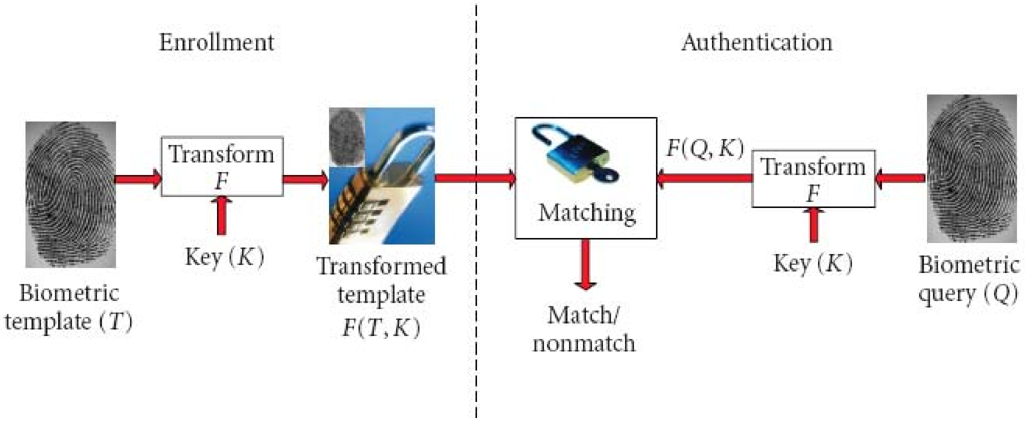

Figure 2.10 Basic Design of Feature Transformation Schemes (Jain, Nandakumar, & Nagar, 2008, p. 8)

2.4.1.1. Experiments Related Feature Transformation Schemes for Fingerprints

Jin, Ling and Goh (2004) developed a two-factor protection scheme called BioHashing. The fingerprint features were transformed using wavelet Fourier-Mellin transform. An inner product was calculated between the transformed features and a pseudo-random sequence stored on a token. The product was discretetized to get a user-specific code that was publically stored. During verification, the process was repeated with the new features and the user code obtained was compared with the one stored for a match. An experiment was performed on the FVC 2002 database, with high performance rates for more bits in the user compact code.

Ratha, Chikkerur, Connell and Bolle (2007) proposed and analyzed three schemes based on the use of non-invertible transforms for fingerprint templates. The transformation functions were Cartesian, polar and functional. A pre-processing step for the transformation included the registration of the minutiae points so that they could be measured with respect to the same coordinate system. Identifying singularities with the help of parabolic and triangular symmetries associated with them and using the singularities to express minutiae positions did this. The Cartesian and polar transformations shifted each cell of the minutiae space to new positions by using rectangular and shell based tessellations respectively. The functional transformation used a mixture of Gaussians and random charge distributions in surface folding functions. An experiment was performed using 188 fingerprints from the IBM-99 optical database, with better performance results for the functional and polar transforms.

2.4.2. Biometric Cryptosystems

Biometric cryptosystems represent another family of biometric protection schemes. In biometric cryptosystems an asymmetry is introduced with the help of a cryptographic construct. A key is associated with the template to generate the secure reference. During authentication, the new raw data is used in association with the reference in the successful recovery of the cryptographic key. Fig 2.11 shows a basic design of a biometric cryptosystem.

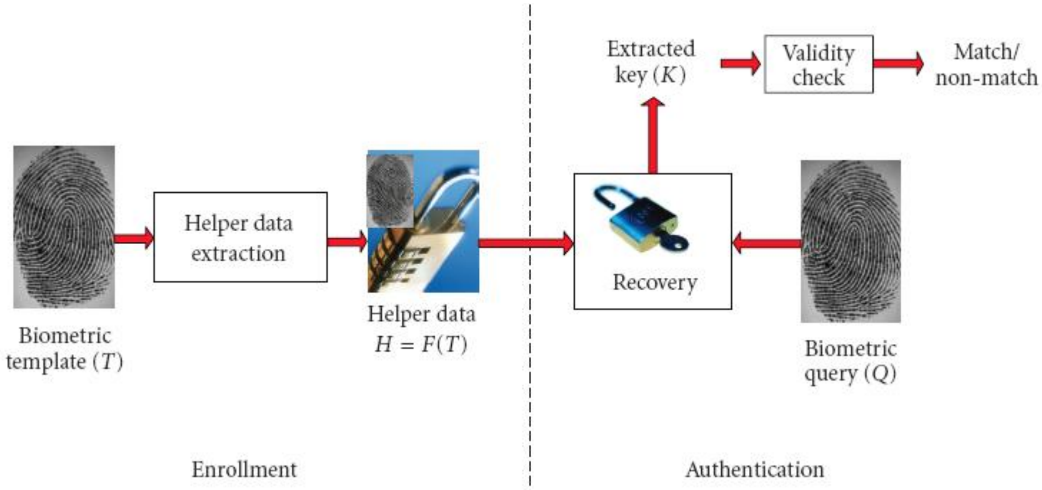

Figure 2.11 Basic Design of Biometric Cryptosystems (Jain, Nandakumar, & Nagar, 2008, p. 9)

2.4.2.1. <u>Experiments Related Biometric Cryptosystem Schemes for Fingerprints</u>

Uludag, Pankanti and Jain (2005) developed a scheme for fingerprints based on the fuzzy vault construct developed by Juels and Sudan (2002). A secret was appended with parity bits and used to derive coefficients for a polynomial. The template features were evaluated on this polynomial to obtain a set of genuine points. This set of genuine points was combined with another set that consisted of random chaff points that did not fall on the polynomial. This composite list of points was shuffled to create the vault. The vault data was publically stored. During authentication, a new set of template features was used to reconstruct the polynomial given the vault data. If the original feature and the new feature set matched, the initial secret was revealed. An experiment was performed with the IBM fingerprint database consisting of a 100 images to obtain a false accept rate of 0%.

Draper, Khisti, Martinian, Vetro and Yedidia (2007) developed a scheme for fingerprints based on distributed source coding techniques implemented using graph-based codes. A statistical model was used to link the enrollment and probe minutiae vectors in terms of a factor graph. The enrollment vector was then compressed as a syndrome using low-density parity codes through a modulo sum function. A belief propagation approach was then used to decode the enrollment vector using the syndrome and the factor graphs. An experiment was performed on a 1000 fingerprint images from the MELCO database with moderate performance results and the conclusion that the scheme was not perfectly secure.

2.4.3. The TURBINE Project

The Turbine project is a major effort by important players in the fields of biometrics and cryptography to produce privacy-enhancing technologies by combining secure automatic user verification using electronic fingerprint authentication and reliable protection of biometric data with the help of advanced cryptographic technologies (Delvaux et al., 2008, p. 1063). The European Union through its 7th framework programme for research and technology development funds the work done under this initiative.

2.4.3.1. Motivation

The primary motivations for the project stemmed from the privacy concerns associated with the use of fingerprints for ID management. Delvaux et al. (2008, pp. 1063-1064) investigated this problem with regard to typical application scenarios for biometric fingerprint recognition in eFinance and eGovernment market sectors. They found that there was limited trust between an individual and his application service providers. The individual was very reluctant to either allow the provider to store the fingerprint in a database or transmit it each time to the provider for an electronic transaction. The existing ID management systems based on fingerprint biometrics were limited in their capability to provide a solution to this issue without a tradeoff on the flexibility of use and fingerprint data security (Busch, 2008, p. 4; Delvaux et al., 2008, p. 1063).

2.4.3.2. Objectives

The primary objective for the project is to provide significant advances over currently used ID management technologies by eliminating the perception that privacy and security are in a zero sum game and showing that they can coexist simultaneously (Delvaux et al., 2008, p. 1066). According to Katholieke Universiteit Leuven (2008, p. 7) this would be achieved through many tasks, some of which are given below:

1. To obtain non-invertible and protected unique bit strings as Pseudo Identities (PI) for enrollment and verification in the ID management system and provide capability to regenerate and revoke independent PI’s based on the same fingerprint.
2. To assess applicability and scalability of the ID system to large populations through detailed 1:1 verification evaluations using PI’s against very large public and private fingerprint databases.

3. To study the process for generation and release of specifications required by vendors of ID management systems for interoperability and to contribute to the international standards (ISO/IEC JTC1 SC 27 WD 24745) related to the project work.

2.4.3.3. Pseudo Identities, Auxiliary Data and Protected Templates

According to Delvaux et al. (2008, p. 1064), for every application that an individual uses, sensitive information such as biometric samples or templates are required to be processed so as to create unique biometric references. The biometric references do not reveal any information that allows retrieval of the original biometric measurement data, biometric template or true identity of the owner by any person other than the enrolled user. These biometric references are binary identity verification strings that either replace or combine with an individual’s actual physical identity and are known as PI.

According to Gjøvik University College (2010, p. 11), Auxiliary Data (AD) is additional information generated during the creation of PI that serves different purposes depending on the methods and algorithms employed. This may include:

1. Generation of multiple PI’s for the same person and same application that are distinct from each other in order to provide renewability.
2. Generation of distinct PI’s across different applications to prevent database cross matching and linking.
3. Generation of distinct PI’s for different people that have similar biometric characteristics to prevent impersonation.
4. Optimized verification performance through individualized performance.

The AD and the PI together provide a high level implementation of the protected template.

2.5. Previous Work Related to Template Splitting

Baltatu et al. (2004) provided a design for template splitting derived on the secret splitting and sharing paradigm proposed in Shamir (1979) within the scope of biometric templates. They referred to template splitting as a method where a template is taken as an input and two entities called template shares created. Each template share stores only a part of the information from the original template. Given a biometric template T during enrollment, a random number R was generated equal in size to T. From R and T, T1 was calculated as T XORed with R. Then T1 and R referred to the template shares and were kept in different storage locations. During authentication, the two parts were recovered from their storage locations and used to reconstruct T by performing an XOR between R and T1. From the properties of the secret splitting and sharing paradigm as described in Shamir (1979), the design will be information theoretic secure since given R, or T1 alone, neither part can be used on its own to obtain the template T. No experiments were conducted.

Baltatu, D'alessandro and D'amico (2008) expanded on previous work done in Baltatu et al. (2004) and included the template splitting within a biometric verification system. The template splitting was part of the enrollment step of the biometric verification system to store the reference template as its shares. The template shares were signed and enciphered before they were stored. During verification, the reference template shares were deciphered and then used in conjunction with the newly acquired biometric sample to produce a matching result. No experiments were conducted.

The author discussed with Modi (personal communication, December 1, 2009) the possibility of using techniques other than secret splitting and sharing for template splitting in fingerprints. The focus was on fingerprint templates that only contained minutiae information. The intention was to avoid the use of external random information as done by Baltatu et al. (2004), instead working with the information already available in the fingerprint template. The designs that were suggested by Modi (personal communication, December 16, 2009) include:

1. Regular order scheme: Dividing minutiae's across the centre of the fingerprint sample.
2. x scheme: Ordering minutiae's according to their x-coordinates and dividing across the centre of the fingerprint sample.
3. y scheme: Ordering minutiae's according to their y-coordinates and dividing across the centre of the fingerprint sample.

It was investigated whether the use of template splitting in a fingerprint verification system caused degradation in system performance. Experiments were conducted in this regard, with system performance measured in terms of FNMR. The experimental results showed that template splitting was acceptable for use in fingerprint verification, as there was a decrease of only 3-5 % in the system performance when it was included. Among the different methods discussed, the y scheme provided the best results with a top FNMR of 4.64% and the x scheme provided the worst results with a top FNMR of 6.17%.

2.6. Error Rates

Error rates are the basic metrics that can provide quantifiable assessment of performance in biometric systems. They can be classified as decision error rates or matching error rates. Decision error rates are calculated over the number of transactions that are made in the biometric system, while matching error rates are calculated on the number of comparisons made by the matching algorithm in the system (Mansfield & Wayman, 2002, pp. 4-6). There are two types of matching error rates, FMR and FNMR.

FMR is the proportion of zero-effort imposter samples submitted by a user in attempts to match with his/her template stored in the biometric system, but falsely declared by the matching algorithm to match the non-self template with which it is being compared (ISO, 2006). The false declaration is caused because the similarity score returned by the matching algorithm for this comparison is above the decision threshold.

FNMR is the proportion of samples from a user used in genuine attempts that are incorrectly declared by the matching algorithm not to match the template of the same characteristic for the same user when compared (ISO, 2006). The false declaration is caused because the similarity score returned by the matching algorithm for this comparison is below the decision threshold.

CHAPTER 3. METHODOLOGY

The main goal of the chapter is to ensure that the evaluation conducted as part of this thesis would be reliable and repeatable.

3.1. Research Design

This thesis used a quasi-experimental research method. The purpose of the research was to evaluate whether template splitting could be used to prevent a good approximation of the original fingerprint sample from being constructed from the fingerprint template. The experiment conducted had a one-group pretest-posttest design. For the experiment, the independent variable was the fingerprint template; the treatment was the splitting scheme and the dependent variable was the FNMR generated by comparing the reconstructed fingerprint sample with the impressions of the original fingerprint. The control factor included physical management of the system on which the experiment was performed.

3.2. Database Description

A database of fingerprint samples was used in this study. The database was collected as part of the research in Modi (2008). The Crossmatch Verifier LC 300 optical touch sensor was used for fingerprint acquisition at a resolution of 500 dots per inch. 6 fingerprint samples were collected from the index finger of the natural hand for 190 subjects at the West Lafayette campus of Purdue University. The total size of the fingerprint database was 6 x 190 = 1140 samples.

3.3. Data Processing Methodology

3.3.1. Feature Extraction and Matching

The matcher and feature extractor available with Neurotechnology VeriFinger® 6.0 was used in this study. Neurotechnology VeriFinger® 6.0 is a commercially available fingerprint identification technology for use by biometric developers and integrators. It was chosen for this study because of the different capabilities provided as part of the application to work with fingerprint samples.

The Neurotechnology VeriFinger® 6.0 extractor acquired the core point(s) from each of the fingerprint samples. The core point(s) had information on x-coordinate and y-coordinate. In case of multiple cores, the uppermost core or the core with the higher y-coordinate value was chosen.

The Neurotechnology VeriFinger® 6.0 extractor also created the fingerprint templates. Each fingerprint template was obtained from a single fingerprint sample. The fingerprint template was composed of minutiae features only. The fingerprint template provided minutiae information according to the minutiae standard defined in ISO (2005). Each minutia had information on x-coordinate, y-coordinate in pixels and orientation in units of 360/256 or 1.40 degrees, with the origin of the coordinate system defined in the upper left corner of the fingerprint sample.

The Neurotechnology VeriFinger® 6.0 matcher compared VeriFinger fingerprint templates to generate a match score. The match score generated was a ratio variable that only had non-negative numbers.

3.3.2. Fingerprint Template Splitting

The fingerprint template was split using the y scheme. This scheme was chosen because it gave the best performance among all the different methods analyzed by the author and Dr. Modi (Refer to section 2.6).

3.3.2.1. Fingerprint Template Splitting Scheme

The representation for minutiae, core, fingerprint template and template shares is first outlined here for clarity.

1. The minutiae is a 3 - tuple M = [x, y, θ], where x, y values are the coordinates and θ is the orientation. M (y) refers to the y-coordinate of the minutiae.
2. The core is a 2-tuple C = [x, y] where x and y values are the coordinates. C (y) refers to the y-coordinate of the core point.
3. The fingerprint template is an n-tuple $T = [M_1, M_2 \ldots M_x]$ where M_i is the i^{th} minutiae and x is the total number of minutiae in T.
4. The template shares $T_{1/2}$ and $T_{2/2}$ are subsets of T with the property: $|T_{1/2}| + |T_{2/2}| = |T|$ and $T_{1/2} \cap T_{2/2} = \emptyset$.

The pseudo code for the template splitting scheme is now presented here:
Initialization: Set $T_{1/2}$ and $T_{2/2}$ so that both are empty. Order T such that if $M_i (y) > M_j (y)$ then $M_i > M_j$.

```
For (each $M_i$ in T)
{
If ($M_i (y) > C (y)$) then
        $M_i$ in $T_{1/2}$
Else
        $M_i$ in $T_{2/2}$
}
```

3.4. Data Analysis Methodology

3.4.1. Technique

The analysis was started with a simulation for a masquerade attack without a template splitting scheme being used at the time of enrollment for a fingerprint sample. The fingerprint template stored at the time of enrollment was compromised and used in a sample reconstruction process to create an approximate sample. The approximate sample was matched with all impressions of the original fingerprint. This yielded a set of genuine comparison scores and a FNMR calculated from it. This FNMR served as the baseline.

Another simulation was run for a masquerade attack, this time with a template splitting scheme being included at the time of enrollment for a fingerprint sample. One of the template shares created at the time of enrollment was compromised and used in a sample reconstruction process to create an approximate sample. The approximate sample was matched with all impressions of the original fingerprint. This yielded another set of genuine comparison scores and another FNMR calculated from it.

The first and the second FNMR were evaluated statistically using the chi-square test for homogeneity of proportions.

3.4.2. Reconstruction of Sample from Minutiae Points

The sample reconstruction process used was taken from Feng and Jain (2009). The work was described in section 2.3.3.1. The idea is to model a fingerprint image as a two-dimensional FM signal inferred from the minutiae information on x-coordinate, y-coordinate and orientation that is available in the template. The process is shown in Fig 3.1. The author did not have access to the work done by J. Feng and A.K. Jain to implement the reconstruction process, so an attempt was made to recreate it.

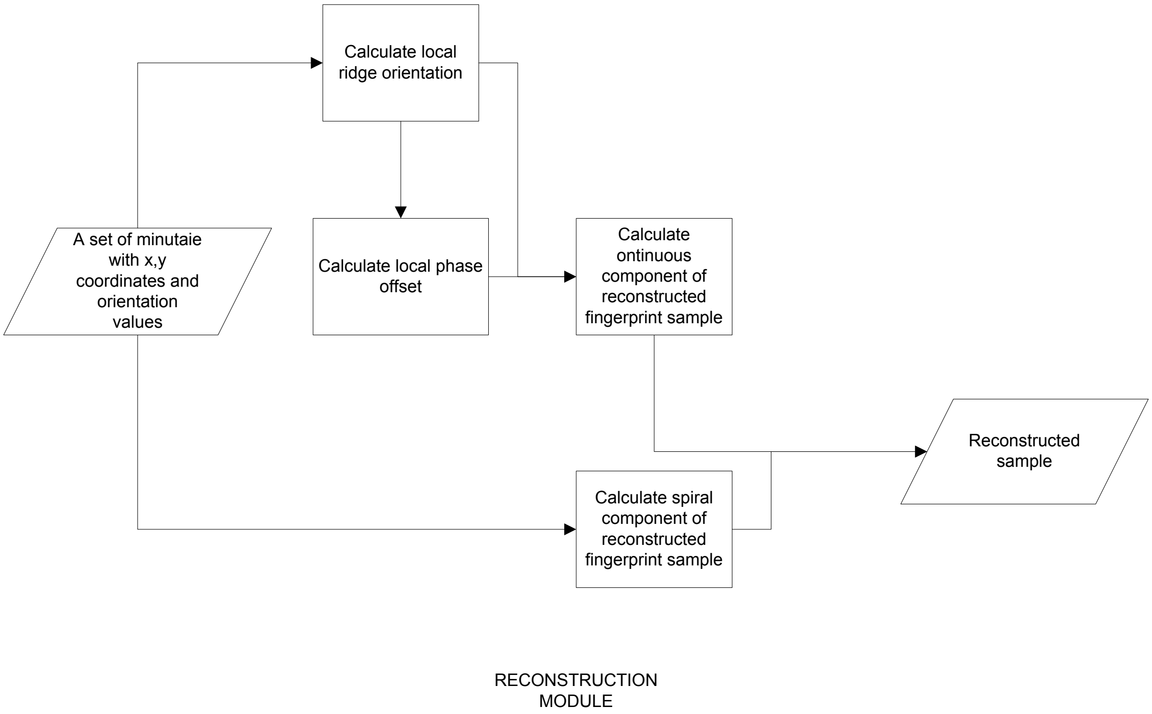
Calculate local ridge orientation
A set of minutaie with x,y coordinates and orientation values
Calculate local phase offset
Calculate ontinuous component of reconstructed fingerprint sample
Reconstructed sample
Calculate spiral component of reconstructed fingerprint sample
RECONSTRUCTION MODULE

Figure 3.1 Sample Reconstruction Process (Feng & Jain, 2009)

3.4.3. Attack Simulation without Template Splitting

There were three processes that were part of this simulation: enrollment, attack and verification. All three processes are shown in Fig 3.2.

In the enrollment process, the enrollment template was generated for a user from a fingerprint sample by using the Neurotechnology VeriFinger® 6.0 extractor and stored in one of the four possible storage locations that had been previously discussed in section 2.1.2. After enrollment, the attack process was initiated. A malicious attacker compromised the storage location and the enrollment template was acquired. The compromised template was then used in congruence with the reconstruction module to obtain a reconstruction sample as described in 3.4.2. During verification, this reconstructed sample was presented and using the Neurotechnology VeriFinger® 6.0 extractor a template was acquired. A test template was extracted from another fingerprint sample for the user by using the Neurotechnology VeriFinger® 6.0 extractor. The test template was then compared with the template acquired from the reconstructed sample by using the Neurotechnology VeriFinger® 6.0 matcher to generate a match score.

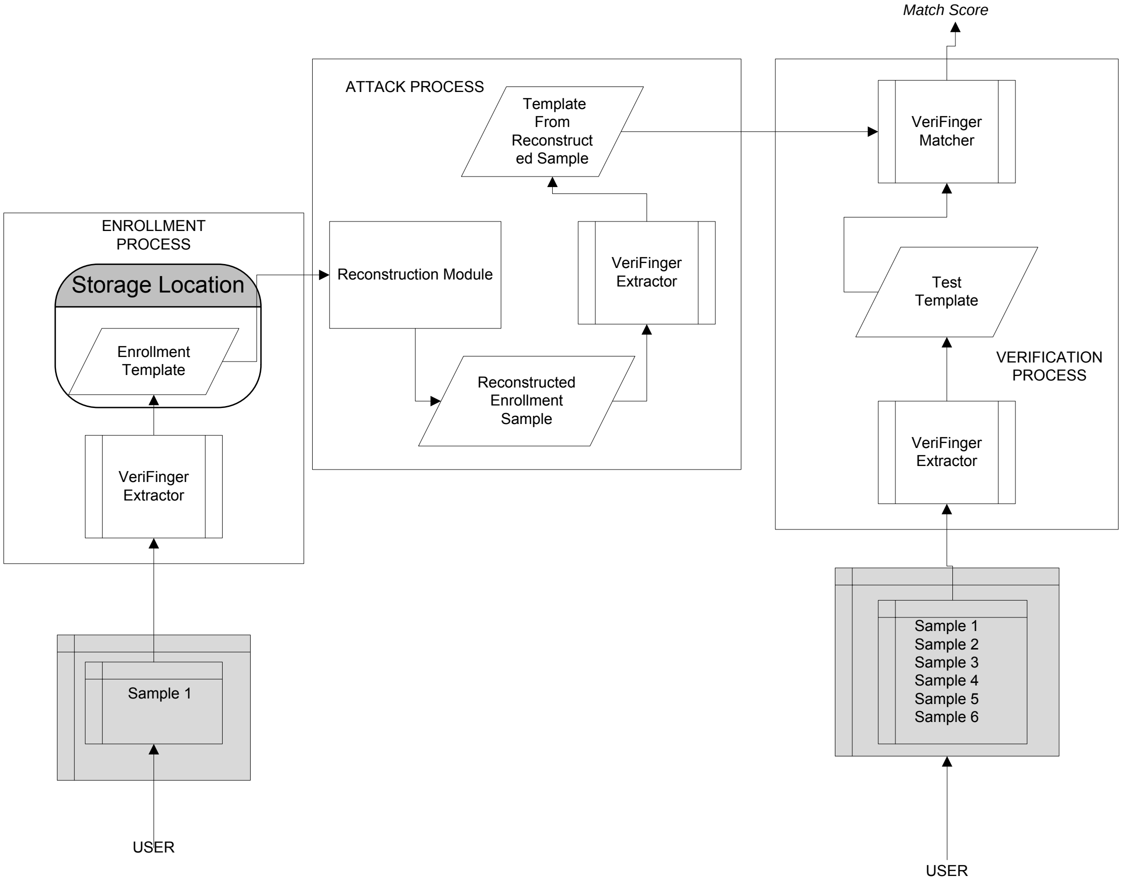

Figure 3.2 Simulation of Masquerade Attack without Template Splitting

For each user, there was one enrollment template and six test templates. The enrollment template was obtained only from the first sample of a user's fingerprint. The test templates were obtained individually from all six samples for the user's fingerprint. As a result there were 6 genuine comparisons for each user, and 6 corresponding genuine comparison match scores.

By repeating the attack simulation for every user in the database, a total of 190 * 6 =1140 genuine comparisons and corresponding genuine comparison match scores were obtained. This set of genuine comparison match scores was referred to as S_1.

3.4.4. Attack Simulation with Template Splitting

There were three processes that were part of this attack system: enrollment, attack and verification. All three processes are shown in Fig 3.3.

In the enrollment process, the enrollment template was generated for a user from a fingerprint sample by using the Neurotechnology VeriFinger® 6.0 extractor. It was then split using the fingerprint template splitting scheme described in 3.3.2.1. The two-enrollment template shares obtained were stored at two physically separate storage locations. These can be any of the four possible storage locations that had been previously discussed in section 2.1.2. After enrollment, the attack process was initiated. A malicious user compromised the storage location 2 and the enrollment template share 2 was acquired. Note that the use of a specific storage location here is for exemplification, and it could be either of the two storage locations shown in Fig.3.3. The primary condition set here is that only a single enrollment share is compromised. The compromised share was then used in congruence with the reconstruction module to obtain a reconstruction sample as described in 3.4.2. During verification, this reconstructed sample was presented and using the Neurotechnology VeriFinger® 6.0 extractor, a template was acquired. A test template was extracted from a fingerprint sample for the user by using the Neurotechnology VeriFinger® 6.0 Extractor. The test template was then compared with the template acquired from the reconstructed sample using the Neurotechnology VeriFinger® 6.0 matcher to generate a match score.

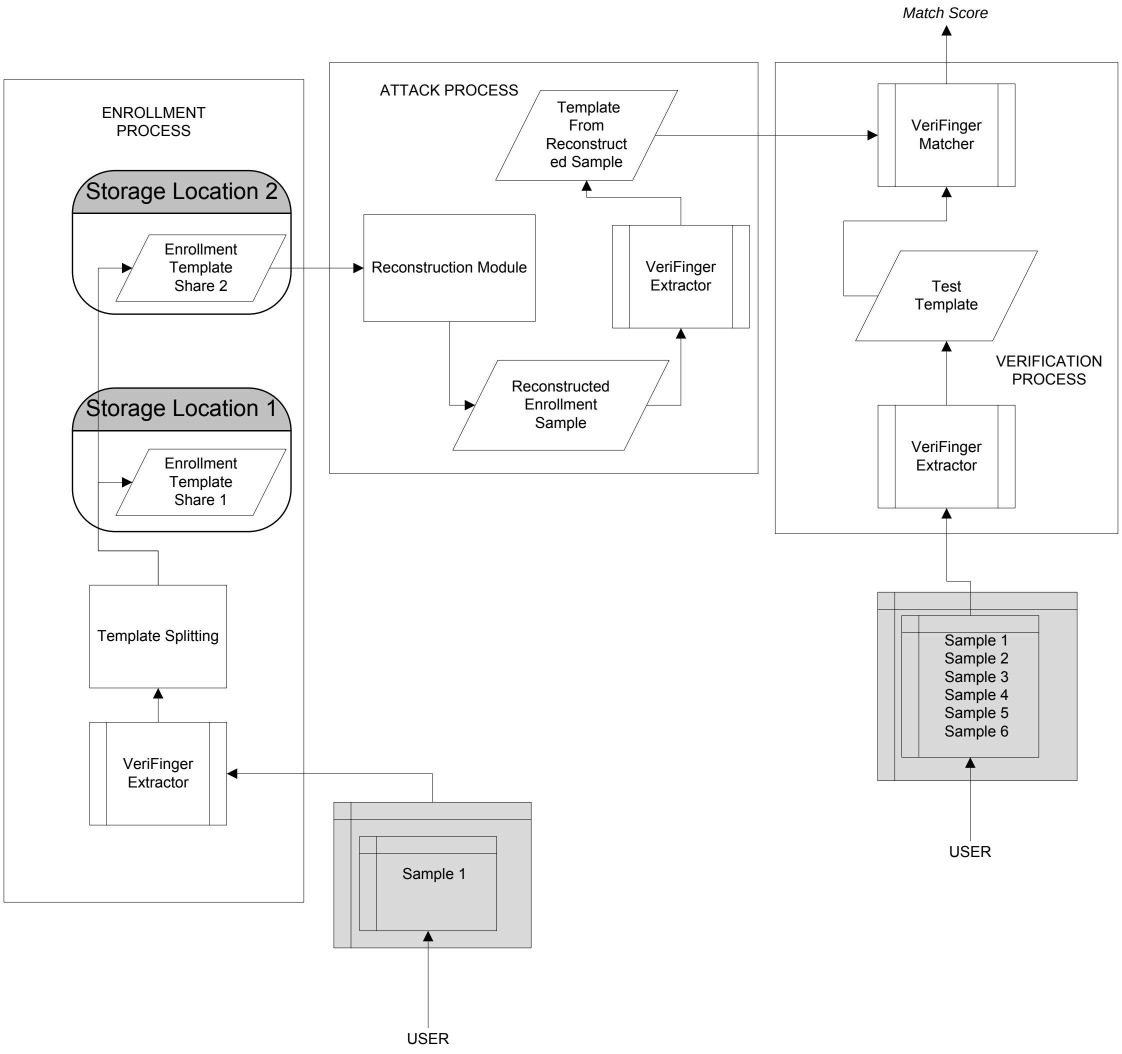

Figure 3.3 Simulation of Masquerade Attack with Template Splitting

For each user, there was one enrollment template and six test templates. The enrollment template was obtained only from the first sample of a user's fingerprint. The test templates were obtained individually from all six samples for user's fingerprint. As a result there were 6 genuine comparisons for each user, and 6 corresponding genuine comparison match scores. By repeating the attack process for every user in the database, a total of 190 * 6 =1140 genuine comparisons and corresponding genuine comparison match scores were obtained. This set of genuine comparison match scores was referred to as S_2.

3.4.5. Calculation of Matching Error Rates

Two sets of genuine comparison match scores S_1 and S_2 were obtained as described in section 3.4.3 and 3.4.4. The FNMR for each set was calculated as given in Eq. 3.1.

$FNMR_1$ = (Number of match scores in S_1 with value < t) /(total number of scores in S_2). Eq. 3.1

$FNMR_2$ = (Number of match scores in S_2 with value < t) /(total number of scores in S_2).

Here t is the matching threshold for decision. According to Neurotechnology (2004, p. 75), the matching threshold is linked to the FMR of the matcher. The higher the value of the threshold, the lower is the FMR and the greater is the FNMR. For the Neurotechnology VeriFinger® 6.0 matcher, matching threshold values for FMR between 1% and 0.001% are recommended to be used since they can be accurately calculated. For this study, the author used a threshold corresponding to a fixed FMR of 0.01% for decision or a matching score of 48.

3.4.6. Statistical Evaluation

The chi-square test for homogeneity is used when r samples are characterized on a single dimension with c categories. The test checks whether the r samples are homogenous with respect to the proportion of observations in each of the c categories (Sheskin, 2004, pp. 493-494). According to Sheskin (2004, p. 494), the test is based on the following assumptions:

1. The r samples should be random, each consisting of n independent observations.

2. The number of samples r and the number of categories c should be greater than or equal to two.
3. The expected frequency of observations for each category in a random sample should be five or greater.

In this study, the chi-square test was used to evaluate the FNMR's. The categories for the observations were match and non-match. The objective of the test was to examine whether the difference between $FNMR_1$ and $FNMR_2$ was statistically significant. A set of null and alternate hypotheses had been formulated for the chi square test. This is given in Eq. 3.2.

H_0: $FNMR_1 = FNMR_2$ Eq. 3.2

H_a: $FNMR_1 \neq FNMR_2$

The chi-square test statistic X^2 was calculated as given in Eq. 3.3.

$X^2 = \Sigma_n (O - E)^2 / E$ Eq. 3.3

O = observed frequency of non matches

E = theoretical frequency of non matches

n: number of categories = 2

In order to evaluate X^2, a chi-square distribution table was used. The chi-square distribution table lists critical values χ^2 in relation to a predetermined significance level and the number of degrees of freedom (Sheskin, 2004, p. 164). In this study, the degrees of freedom were 1. The significance levels of 0.05 and 0.01 were chosen. The critical values $\chi^2_{0.05}$ and $\chi^2_{0.01}$ corresponding to these significance levels and degrees of freedom are 3.841 and 6.635 respectively.

If $X^2 < 3.841$, the null hypothesis that $FNMR_1$ and $FNMR_2$ are not different was supported at the 0.05 significance level. If $3.841 < X^2 < 6.635$, the null hypothesis was rejected at the 0.05 significance level but supported at the 0.01 significance level. If $X^2 > 6.635$, the null hypothesis was rejected at both 0.05 and 0.01 significance levels.

3.5. Summary

This chapter has outlined the research design for the study and explained in detail the data processing and data analysis methodology that was employed by the author. Both the data processing and data analysis methodology were based on previous research related to attacks on fingerprint recognition systems. The section on data analysis also covered the hypotheses that were tested in order to identify the effects of template splitting on fingerprint sample reconstruction.

CHAPTER 4. ANALYSIS AND RESULTS

This chapter outlines the results obtained by performing the data analysis methodologies and statistical tests on the fingerprint samples acquired from a single optical sensor as described in chapter 3. The discussion is in three parts: first the reconstruction of fingerprint samples from minutiae, second the generation of FNMR error rates by conducting attack simulations with and without template splitting, and finally the statistical evaluation of generated FNMR's using the chi-square distribution test for homogeneity of proportions.

4.1. Fingerprint Reconstruction

The reconstruction process was outlined in Sections 2.3.3.1 and 3.4.2. A visual representation of the process is provided in Fig 4.1. The steps are briefly summarized here.

1. To mark the region within the image space that corresponded to the reconstruction sample, a convex hull was obtained from the minutiae points and dilated using a set of disk shaped structuring elements of size 8x8 pixels. This dilated hull served as the foreground mask. The foreground mask is given in the upper left image of Fig 4.1.
2. For the foreground mask, an orientation map was defined. Each value in the map corresponded to the local ridge orientation of a foreground block of size 8x8 pixels. The local ridge orientation for a single foreground block was calculated as the weighted sum of the angular components of the closest minutiae in each of the eight sectors around the block. The orientation map is given in the upper right image of Fig 4.1.

3. Using the orientation map, the continuous phase was constructed by using piecewise linear planes at each foreground block. In order to compensate for difference in phase between neighboring blocks in the foreground mask, a block-offset value was calculated at the boundary of each set of adjoining blocks.
4. The coordinate values of the minutiae points were used to obtain the spiral phase for the foreground blocks within a set of arctangent mathematical functions. The spiral phase is given in the lower left image of Fig 4.1.
5. The continuous and spiral phase where then combined to obtain the reconstructed image. The final reconstructed fingerprint is given in the lower right image of Fig 4.1.

For a more detailed explanation of the mathematical formulations used in the reconstruction process, one can refer to Feng and Jain (2009).

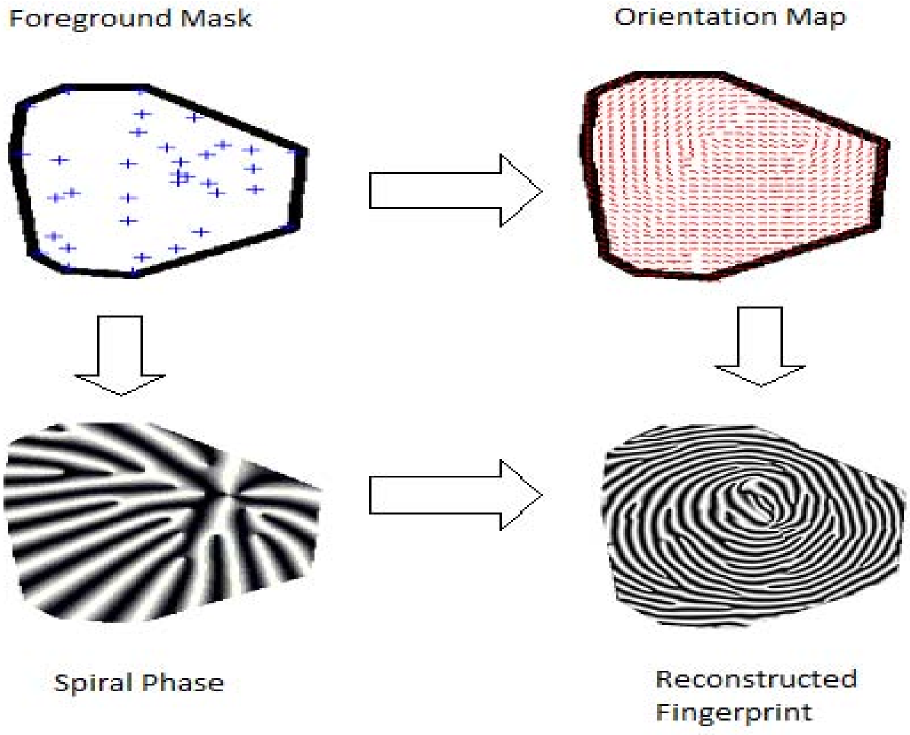

Figure 4.1 The Reconstruction Process for a Set of Minutiae

The author implemented the reconstruction process using the Mathworks Matlab ® 7.7.0.471 software. The implementation was conducted with the assistance of Feng (personal communications, 1-11 June, 2010) who provided an insight into some of the intricate details of the reconstruction process as performed in Feng and Jain (2009). A sample of the reconstruction code is provided in Appendix A.

4.2. Attack Simulations with and without Template Splitting for Generation of FNMR

The processes involved in the attack simulations with and without template splitting were described in sections 3.4.3 and 3.4.4. The steps conducted by the author as part of the experiment are outlined here:

1. The Neurotechnology VeriFinger® 6.0 Extractor was used in an attempt to extract the core from all 1140 fingerprint samples in the fingerprint database. For 18 users in the fingerprint database, no core points could be detected in any of their 6 fingerprint impressions. The samples corresponding to these users were disregarded, resulting in 1032 fingerprint samples being used in the experiment.
2. The Neurotechnology VeriFinger® 6.0 Extractor was then used in an attempt to extract the fingerprint template from the 1032 fingerprint samples, each fingerprint template from a single fingerprint sample. No templates could be extracted for 8 fingerprint samples.
3. The 1024 fingerprint templates where then processed using an implementation of the fingerprint template splitting scheme described in 3.4.2.1. The implementation was written in the C programming language and the code is included as part of Appendix B. The processing resulted in 2048 template shares, 2 template shares for each of the 1024 fingerprint templates.
4. In the attack simulation without template splitting, the fingerprint template corresponding to the user's first sample was used for the reconstruction

process described in section 4.1, resulting in 172 reconstructed fingerprint samples. The reconstructed sample for each user was matched with the corresponding impressions of the user by using the Neurotechnology VeriFinger® 6.0 Matcher.

There were 1024 matching comparisons at a fixed FMR of 0.01% or at a match score threshold of 48, leading to a FNMR of 13.86%.

5. In the attack simulation with template splitting, the second share of the fingerprint template corresponding to the user's first sample was used for the reconstruction process described in section 4.1, resulting in 172 reconstructed fingerprint samples. The reconstructed sample for each user was matched with the corresponding impressions of the user by using the Neurotechnology VeriFinger® 6.0 Matcher.

   There were 1024 matching comparisons at a fixed FMR of 0.01% or at a match score threshold of 48, leading to a FNMR of 38.96%.

A summary of the experimental results is provided in Table 4.1.

Table 4.1 Results of Attack Simulations with and without Template Splitting

Simulation	Total number of genuine match comparisons	Number of false non matches observed in the genuine match comparisons	FNMR at fixed FMR of 0.01%
Attack simulation without template splitting	1024	142	13.86
Attack simulation with template splitting	1024	399	38.96

4.3. False Non Match Overlap between Attack Simulations with and without Template Splitting

The false non matches obtained after conducting the attack simulations with and without template splitting was evaluated to check whether there was any overlap between the two sets. It was observed that out of the 142 false non matches for the attack simulation without template splitting, 21 false non matches were not present in the set for the attack simulation with template splitting. The result was a false non match overlap of 85.21%. A visual analysis of the reconstructed samples that produced these 21 false non matches was done by the author. It showed that the use of template splitting within the simulation excluded some of the deformed regions within the reconstructed sample and caused a genuine match to be recorded.

4.4. Statistical Evaluation of Generated FNMR

The test of homogeneity of proportions using the chi square distribution was performed for the FNMR generated using Neurotechnology VeriFinger® 6.0. This was previously described in section 3.4.6. The significance levels of 0.05 and 0.01 were used. The test was conducted for the hypothesis stated in Eq. 3.2 that is restated in Eq. 4.1.

$$H_0: FNMR_1 = FNMR_2 \qquad \text{Eq. 4.1}$$

$$H_a: FNMR_1 \neq FNMR_2$$

The 2x2 contingency table was created by the author and is shown in Table 4.2.

Table 4.2 Contingency Table for Chi-Square Distribution Test

Contingency Table	Number of genuine matches	Number of false non matches	Row Sums
Simulation without template splitting	882	142	1024
Simulation with template splitting	625	399	1024
Column Sums	1507	541	2048

The chi-square test statistic X^2 was calculated as given in Eq. 3.3 which is restated in Eq. 4.2.

$X^2 = \Sigma_n (O - E)^2 /E$ Eq. 4.2

O = observed frequency of non matches

E = theoretical frequency of non matches

n: number of categories = 2

Using Table 4.2,

$O_{11} = 882$, $O_{21} = 625$, $O_{12} = 142$, $O_{22} = 399$

$E_{11} = E_{21} = 753.95$, $E_{12} = E_{22} = 270.50$

$$X^2 = (882 – 753.95)^2 / 753.95 + (142 – 270.50)^2 / 270.50 + (625 – 753.95)^2 / 753.95 + (399 – 270.50)^2 / 270.50$$

$$= 21.74 + 61.04 + 22.05 + 61.04$$

$$= 165.87$$

The critical values $\chi^2_{0.05}$ and $\chi^2_{0.01}$ corresponding to the significance levels of 0.05 and 0.01 were 3.841 and 6.635 respectively. Since $X^2 > 3.841$ and $X^2 > 6.635$, the null hypothesis was rejected at both 0.05 and 0.01 significance levels. Hence, the difference between the FNMR generated from the attack simulation without template splitting and the FNMR generated from the attack simulation with template splitting was significant.

CHAPTER 5. CONCLUSIONS AND RECOMMENDATIONS

This thesis has described the use of fingerprint template splitting to prevent sample reconstruction from a minutiae based fingerprint template. In this chapter, relevant conclusions based on the work done in this thesis are presented and recommendations made for possible future work.

5.1. Conclusions

The use of a fingerprint template splitting scheme resulted in an increase in the FNMR when performing sample reconstruction within a verification setting. The increase in FNMR was statistically significant, which implies that the inclusion of template splitting causes a decrease in the capability of the sample reconstruction of fingerprints to affect fingerprint systems.

An important consideration here is the variability in system performance associated with the inclusion of template splitting within a fingerprint recognition system. It was previously shown that using the y-scheme for fingerprint template splitting causes a nominal decrease of approximately 3% in system performance (Refer to section 2.5). A 25% decrease in the capability of sample reconstruction with a corresponding decrease in system performance of about 3% appears to be a reasonable tradeoff between security and performance for a fingerprint verification system.

This suggests the viability of using fingerprint template splitting schemes in real world implementation of fingerprint systems to prevent attacks by malicious users. Any inclusion of template splitting within a real world fingerprint system would require the following two security assumptions to hold true:

1. Implicit trust on the software/hardware module implementing the template splitting scheme.
2. Availability of mechanisms to establish a trusted computing environment for the module to function within and communicate with the relevant subsystems of the fingerprint recognition system.

5.2. Recommendations and Future Work

There are different possible extensions to the work done in this thesis.

1. Template splitting could be used for distributed matching-based verification. Fig 5.1 shows a generic outline of such a system. The distributed matching could be evaluated in its capability to improve the security of a user's template against malicious users and the corresponding variation in system performance.

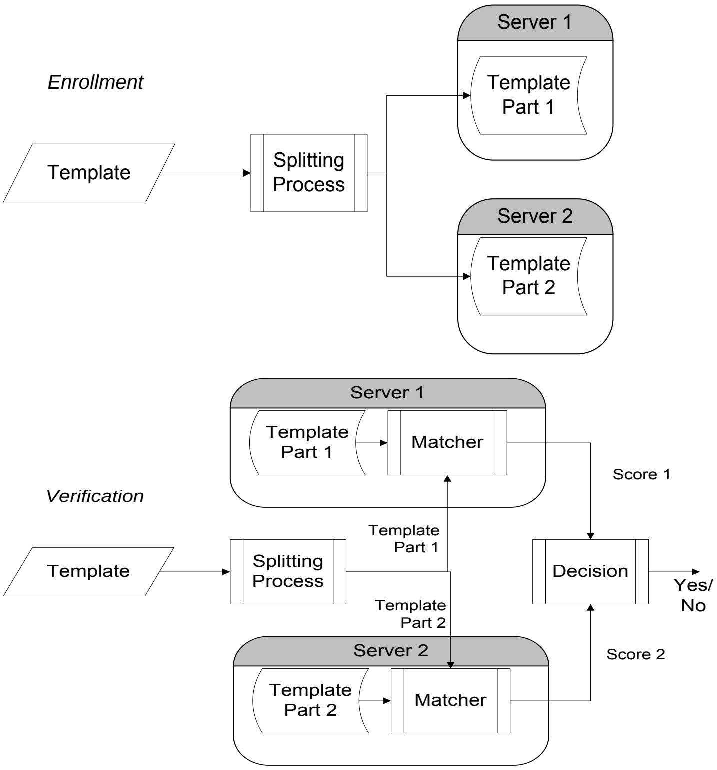

Figure 5.1 Distributed Storage and Matching Verification System (Modi, personal communication, October 15, 2009)

2. The analysis methodology could be modified to test fingerprint template splitting for FMR as part of an identification setting.
3. Other reconstruction mechanisms such as those given in Cappelli et al. (2006) and Ross et al. (2005) and template splitting mechanisms such as the regular order and x schemes could be used and cross evaluated. The template splitting schemes could be modified to create and use more than two template shares.
4. The methodology could be recreated to experiment with other databases such as those made publically available by NIST or FVC to evaluate whether template splitting is consistent in its effects on the capability of sample reconstruction to affect fingerprint systems..

LIST OF REFERENCES

LIST OF REFERENCES

Ad Hoc Group on Biometrics in E-Authentication. (2007). *Study report on biometrics and e-authentication* (Study Report No. 07-0185). Washington, DC: InterNational Committee for Information Technology Standards. Retrieved 28th April 2010 from http://www.incits.org/tc_home/m1htm/m1070185rev.pdf.

Adler, A. (2003). Sample images can be independently restored from face recognition template. In *Proceedings of Canadian Conference on Electrical and Computer Engineering* (Vol. 2, pp. 1163–1166).

Adler, A. (2004). Images can be regenerated from quantized biometric match score data. In *Proceedings of Canadian Conference on Electrical and Computer Engineering* (Vol. 1, pp. 469-472).

American Dermatoglyphics Association. (1990). Taking Dermatoglyphic prints: A self-instruction manual. (T. Reed & R. Meier, Eds.) *Supplement to the Newsletter of the American Dermatoglyphics Association*, *9*, 1-45.

Ashbaugh, D. (1991). Ridgeology. *Journal of Forensic Identification, 41*(1), 16-64.

Atmel. (2004). Fingerchip technology for biometric security. *Atmel Applications Journal*, (2), 35- 40. Retrieved 28th April 2010 from www.atmel.com/dyn/resources/prod_documents/fingerchip.pdf.

Babler, W. J. (1991). Embryologic Development of Epidermal Ridges and their Configurations. *Dermatoglyphics: science in transition*, *27*(2), 95–112.

Baltatu, M., D’Alessandro, R., & D’Amico, R. (2004). Toward ubiquitous acceptance of Biometric authentication: Template protection techniques. In *Biometric Authentication*, Lecture Notes in Computer Science (Vol. 3087, pp. 171–183). Heidelberg: Springer Berlin.

Baltatu, M., D'alessandro, R., & D'amico, R. (2008). User Authentication Method Based On The Utilization Of Biometric Identification Techniques And Related Architecture. *U.S. Patent Application Publication*. Available from http://patft.uspto.gov/.

Bazen, A. M., & Gerez, S. H. (2000). Directional field computation for fingerprints based on the principal component analysis of local gradients. In *Proceedings of ProRISC2000, 11th Annual Workshop on Circuits, Systems and Signal Processing*. Veldhoven, The Netherlands.

Bazen, A. M., & Gerez, S. H. (2001). Segmentation of fingerprint images. In *Proceedings of ProRISC2001, 12th Annual Workshop on Circuits, Systems and Signal Processing* (pp. 276- 280). Veldhoven, The Netherlands.

Berry, J., & Stoney, D. (2001). History and Development of Fingerprinting. In H. Lee & R. E. Gaensslen (Eds.), *Advances in fingerprint technology*, Forensic and Police Science Series (2nd ed., pp. 1-41). Boca Raton, FL: CRC Press.

Busch, C. (2008). *TURBINE: background and status ISO standardisation initiative*. Presented at the EBF Biometric Encryption Seminar, Amsterdam. Retrieved 7th May 2010 from http://www.eubiometricsforum.com/pdfs/be/be-busch.pdf.

Cappelli, R., Lumini, A., Maio, D., & Maltoni, D. (1999). Fingerprint classification by directional image partitioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(5), 402–421. doi:10.1109/34.765653.

Cappelli, R., Lumini, A., Maio, D., & Maltoni, D. (2006). Can Fingerprints be Reconstructed from ISO Templates? (pp. 1–6). Presented at the 9th International Conference on Control, Automation, Robotics and Vision. doi:10.1109/ICARCV.2006.345478.

Cappelli, R., Maio, D., Lumini, A., & Maltoni, D. (2007). Fingerprint image reconstruction from standard templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(9), 1489–1503. doi:10.1109/TPAMI.2007.1087.

Clarke, R. (2002). Biometrics' inadequacies and threats, and the need for regulation. Australian National University. Retrieved 28th April 2010 from http://www.rogerclarke.com/dv/biometrics.html.

Cole, S. A. (2004). History of Fingerprint Pattern Recognition. In N. Ratha & R. Bolle (Eds.), *Automatic Fingerprint Recognition Systems* (pp. 1-25). New York: Springer.

Delvaux, N., Chabanne, H., Bringer, J., Kindarji, B., Lindeberg, P., Midgren, J., Breebaart, J., et al. (2008). Pseudo Identities Based on Fingerprint Characteristics. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing.* (pp. 1063–1068). Harbin. doi:10.1109/IIH-MSP.2008.327.

DigitalPersona. (2006). DigitalPersona pro for active directory- Administrator guide. Retrieved 28th April 2010 from http://www.digitalpersona.com/uploadedfiles/support/reference_material/guides/dppro4_0adminguide.pdf

Draper, S. C., Khisti, A., Martinian, E., Vetro, A., & Yedidia, J. S. (2007). Using distributed source coding to secure fingerprint biometrics. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing* (Vol. 2, pp. 129- 132). doi:10.1109/ICASSP.2007.366189.

Feng, J., & Jain, A. (2009). FM model based fingerprint reconstruction from minutiae template. In *Advances in Biometrics*, Lecture Notes in Computer Science (Vol. 5558, pp. 544- 553). Heidelberg: Springer Berlin.

Fujieda, I., Ono, Y., & Sugama, S. (1995). US Patent 5446290. Fingerprint image input device having an image sensor with openings.

Gjøvik University College. (2010). *Research findings for standardisation* (Project Deliverable No. D2.3.3). TURBINE Consortium. Retrieved 7th May 2010 from http://www.turbine-project.eu/dowloads/turbine-guc-d2.3.3-standardisation.r1.pdf.

Galbally, J., Cappelli, R., Lumini, A., Gonzalez-De-Rivera, G., Maltoni, D., Fierrez-Aguilar, J., Ortega-Garcia, J., & Maio, D. (2010). An Evaluation Of Direct Attacks Using Fake Fingers Generated From ISO Templates, *Pattern Recognition Letters*, 31(8), 725-732.

Galbally, J., Cappelli, R., Lumini, A., Maltoni, D., & Fierrez-Aguilar, J. (2008). Fake Fingertip Generation From A Minutiae Template, *In Proceedings of 19th International Conference On Pattern Recognition, Tampa, Florida, USA* (pp.1-4).

Hill, C. J. (2001). *Risk of masquerade arising from the storage of biometrics* (Bachelors thesis, Department of Computer Science). Australian National University. Retrieved 28th April 2010 from http://chris.fornax.net/download/thesis/pdf.

Hong, L., Wan, Y., & Jain, A. (1998). Fingerprint image enhancement: algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 777- 789. doi:10.1109/34.709565.

International Biometrics Group. (2002). *Generating Images from Templates* (Technical Report). I.B.G. White Paper. Available from http://www.biometricgroup.com/reports/public/reports/templates_images.html.

International Biometrics Group. (2009). Biometrics market and industry report 2009-2014. Available from http://www.biometricgroup.com/reports/public/market_report.php.

InterNational Committee for Information Technology Standards. (2004). ANSI/INCITS 378 – Information technology–Finger minutiae format for data interchange (No. ANSI/INCITS 378: 2004). *INCITS*.

International Organization for Standardization. (2005). ISO/IEC 19794-2: Information Technology- Biometric data interchange formats – Part 2: Finger minutiae data (No. ISO/IEC 19794-2(E)). *Geneva:ISO/IEC*.

International Organization for Standardization. (2006). ISO/IEC 19795-1: Information technology - Biometric performance testing and reporting - Part 1: Principles and framework (No. ISO/IEC 19795-1(E)). *Geneva: ISO/IEC*.

International Organization for Standardization. (2008). ISO/IEC JTC1/SC37 Standing document 2 version 10- Harmonized biometric vocabulary (No. SC37N2777). *Geneva: ISO/IEC.*

Jain, A. K., Nandakumar, K., & Nagar, A. (2008). Biometric template security. *EURASIP Journal on Advances in Signal Processing*, 1–17. doi:10.1155/2008/579416.

Jain, A. K., Prabhakar, S., & Pankanti, S. (2002). On The Similarity of Identical Twin
Fingerprints. *Pattern Recognition*, *35*(11), 2653–2663. doi: 10.1016/S0031-3203(01)00218-7.

Jain, A. K., Ratha, N. K., & Lakshmanan, S. (1997). Object detection using Gabor filters. *Pattern Recognition*, *30*(2), 295–309. doi:10.1016/S0031-3203(96)00068-4.

Jain, A. K., Ross, A., & Prabhakar, S. (2004). An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, *14*(1), 4–20. doi:10.1109/TCSVT.2003.818349.

Jin, A. T., Ling, D. N., & Goh, A. (2004). Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recognition*, *37*(11), 2245–2255. doi:10.1016/J.PATCOG.2004.04.011.

Juels, A., & Sudan, M. (2006). A fuzzy vault scheme. *Designs, Codes and Cryptography*, *38*(2), 237–257. doi:10.1007/S10623-005-6343-Z.

Karu, K., & Jain, A. K. (1996). Fingerprint classification. *Pattern Recognition*, *29*(3), 389–404. doi:10.1016/0031-3203(95)00106-9.

Katholieke Universiteit Leuven. (2008). *Services and schemes for multiple trusted identity* (Project Deliverable No. D1.2.1). TURBINE Consortium. Retrieved 6th May 2010 from http://www.turbine-project.eu/dowloads/turbine-kul-d121-general_scheme-r1.0.pdf.

Laufer, B. (2000). History of the Fingerprint System. *The Print*, *16*(2), 1-13. Retrieved 28th April 2010 from http://www.scafo.org/library/160201.html (Originally published in 1912).

Li, Y., Yin, J., Zhu, E., Hu, C., & Chen, H. (2008). Score based biometric template selection (pp. 1–4). Presented at the 19th International Conference on Pattern Recognition, Tampa, FL. doi:10.1109/ICPR.2008.4761116.

Maio, D., & Maltoni, D. (1997). Direct gray-scale minutiae detection in fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(1), 27–40. doi:10.1109/34.566808.

Maio, D., & Maltoni, D. (1998). Ridge-line density estimation in digital images. In *Proceedings of Fourteenth International Conference on Pattern Recognition* (Vol. 1, pp. 534-538). Brisbane, Qld. doi:10.1109/ICPR.1998.711198.

Maltoni, D., & Cappelli, R. (2008). Fingerprint Recognition. In A. K. Jain, P. Flynn, & A. A. Ross (Eds.), *Handbook of Biometrics* (pp. 23-42). New York, NY: Springer US.

Maltoni, D., Maio, D., Jain, A. K., & Prabhakar, S. (2009). *Handbook of Fingerprint Recognition* (2nd ed.). London: Springer-Verlag London Limited.

Mansfield, A., & Wayman, J. (2002). *Best practice standards for testing and reporting on biometric device performance* (NPL Report No. CMSC 14/02). Queens Road, Teddington, Middlesex: National Physical Laboratory. Retrieved 28th April 2010 from http://www.cesg.gov.uk/policy_technologies/biometrics/media/bestpractice.pdf.

Mehtre, B. M., Murthy, N. N., Kapoor, S., & Chatterjee, B. (1987). Segmentation of fingerprint images using the directional image. *Pattern Recognition*, *20*(4), 429–435. doi:10.1016/0031-3203(87)90069-0.

Modi, S. K. (2008). *Analysis of fingerprint sensor interoperability on system performance* (PHD Dissertation, Dept. of Technology). Purdue University. Retrieved 28th April 2010 from https://www.cerias.purdue.edu/assets/pdf/bibtex_archive/2008-22.pdf.

Morland, N. (1950). *An Outline of Scientific Criminology* (1st ed.). New York: Philosophical Library, Inc.

National Institute of Science and Technology. (2007). ANSI/NIST–ITL 1: Data format for the interchange of fingerprint, facial, and other biometric information (No. ANSI/NIST–ITL 1: 2007). (M. McCabe & E. Newton, Eds.) *NIST Special Publications*, *500*(271).

National Science and Technology Council Subcommittee on Biometrics and Identity Management. (2006). Fingerprint recognition. Retrieved 28th April 2010 from www.biometrics.gov/Documents/FingerprintRec.pdf.

Neurotechnology. (2004). VeriFinger 4.2 SDK Documentation. Retrieved 12th May 2010 from http://www.neurotechnology.com/download/VF_42_SDK.pdf.

Newman, G., & McNally, M. (2005). *Identity theft literature review*. Grant Report for U.S Department of Justice, New York, NY: University of Albany, Rutgers University. Retrieved 7th May 2010 from http://www.idtheft.com/intro.php.

O'Gorman, L., & Xia, X. (2003). Innovations in fingerprint capture devices. *Pattern Recognition*, *36*(2), 361–369. doi:10.1016/S0031-3203(02)00036-5.

Parziale, G. (2007). Touchless Fingerprinting Technology. In N. K. Ratha & Govindaraju, V. (Eds.), *Advances in Biometrics* (pp. 25–48). Springer London.

Ratha, N. K., Chikkerur, S., Connell, J. H., Bolle. (2007). Generating cancelable fingerprint templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(4), 561–572. doi:10.1109/TPAMI.2007.1004.

Ratha, N. K., Connell, J. H., & Bolle, R. M. (2001a). An analysis of minutiae matching strength. In *Audio- and Video-Based Biometric Person Authentication*, Lecture Notes in Computer Science (Vol. 2091, pp. 223–228). Heidelberg: Springer Berlin.

Ratha, N. K., Connell, J. H., & Bolle, R. M. (2001b). Enhancing security and privacy in biometrics-based authentication systems. *IBM systems journal*, *40*(3), 614-633.

Ross, A., Shah, J., & Jain, A. K. (2005). Towards reconstructing fingerprints from minutiae points. *Proc. SPIE, Biometric Technology for Human Identification II*, *5779*, 68–80. doi:10.1.1.69.3846.

Ross, A., Shah, J., & Jain, A. K. (2007). From template to image: reconstructing fingerprints from minutiae points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 544–560. doi:10.1109/TPAMI.2007.1018.

Shamir, A. (1979). How to share a secret. *Communications of the ACM*, *22*(11), 612–613.

Sheskin, D. (2004). *Handbook of parametric and nonparametric statistical procedures* (3rd ed.). Boca Raton, FL: CRC Press.

Soutar, C. (2002). Biometric system security. Bioscrypt Inc (Now, L-1 Enterprise Access Solutions). Retrieved from http://www.comp.hkbu.edu.hk/~ycfeng/project/biometric%20system%20security.pdf.

Soutar, C., Gilroy, R., & Stoianov, A. (1999). Biometric system performance and security. Presented at the First IEEE Workshop on Automatic Identification Advanced Technologies.

Uludag, U., Pankanti, S., & Jain, A. K. (2005). Fuzzy Vault for Fingerprints. In *Audio-and Video-Based Biometric Person Authentication*, Lecture Notes in Computer Science (Vol. 3546, pp. 310–319). Heidelberg: Springer Berlin.

Uludag, U., Ross, A., & Jain, A. K. (2004). Biometric template selection and update: a case study in fingerprints. *Pattern Recognition*, *37*(7), 1533–1542. doi:10.1016/J.PATCOG.2003.11.012.

UPEK. (2008). UPEK solutions- Chipsets and sensors. Retrieved 28th April 2010 from http://www.upek.com/solutions/pc_and_networking/chipsets_sensors.asp.

Young, N. D., Harkin, G., Bunn, R. M., McCulloch, D. J., Wilks, R. W., & Knapp, A. G. (1997). Novel fingerprint scanning arrays using polysilicon TFT's on glass and polymer substrates. *IEEE Electron Device Letters*, *18*(1), 19–20. doi:0.1109/55.553063.

APPENDICES

Appendix A.

```
function[I]= ReconstructPhase(minutiaeFileName)

% get minutiae information: x,y coordinates, and direction
[x y theta]= getMinutiae(minutiaeFileName);

% construct the foreground mask
se = strel('disk',6);
k = convhull (x, y , {'Qt'});
mask = poly2mask(x (k), y (k), 480, 640);
In = imdilate(mask, se);

% get orientation for all blocks of 8x8 pixels within the foreground mask.
% Also return the block centers
[blockX blockY blockO] = FindOrientation (x, y, theta);

% calculate spiral phase
spiral = 0;
[m n] = meshgrid(1:640,1:480);
for i = 1:length(x)
        if(theta(i) <= 180)
                p = 1;
        end
        if(theta(i) > 180)
                p = -1;
        end
spiral = spiral + p * atan2((m-y(i)),(n-x(i)));
end

% calculate continous phase
continous = zeros([480 640]);
for i = 1:60
   for j = 1:80
        [e g]= meshgrid((i-1)* 8+1:1:i * 8, (j-1)* 8+1:1:j * 8);
          continous ((i-1)* 8+1 :i * 8, (j-1)* 8+1 :j * 8) =
        2 * 0.1 * pi * (e.* cos(blockOrientation(i,j)) + g.* sin(blockOrientation(i,j)));
   end
end
% code for resetting the image blocks in order to account for phase offset is not
included as part of this code sample. It can be implemented with the use of a
depth/breadth first search starting with the root at the upper leftmost foreground
block. An auxiliary image will be required to store the final result.
I = cos((continous + spiral). * In);
%----------------------END OF FUNCTION------------------------------------
```

```
% gets x,y coordinates and orientation for minutaie from file and stores in
% matrices
function[x y theta]= getMinutiae(minutiaeFileName)

% open minutiae files to read values and set storage matrices
fid = fopen (minutiaeFileName, 'r');
A = fscanf(fid,'%d%d%d');
x = zeros([fix(length(A)/3) 1])
y = x;  theta = y;  size = 1;

% format minutaie information while reading from file
for i = 1:3:length(A)
        x(size) = A (i);
        y(size) = A (i + 1);
        theta(size) = A (i + 2) * (360/256) * (pi/180);
        size = size + 1;
end

fclose(fid);
%----------------------END OF FUNCTION------------------------------------

% finds orientation values and block centers for each block
function[blockX blockY blockO]= FindOrientation(x, y, theta)

blockX = zeros([60 80]); blockY = blockX; blockO = blockX;

for i = 1:60
   for j = 1:80
       blockX(i,j) = 4 + 8 * (i - 1);
       blockY(i,j) = 4 + 8 * (j - 1);
       blockO(i,j)= OrientationByBlock(x, y, theta, blockX(i,j),blockY(i,j));

   end
end
%----------------------END OF FUNCTION------------------------------------

% finds nearest neighbour for each block and calculates the orientation
function[m]= OrientationByBlock (x, y, theta, a, b)

u = 0; v = 0;
distanceXY = zeros([length(x) 1]); angleXY = distanceXY;

% find angles and distance of minutiae from block centers
for k = 1: length(x)
```

```
          angleXY(k) = atan2((y(k)- b),(x(k)- a)) + pi;
          distanceXY(k) = sqrt ((x(k)- a) ^ 2 + (y(k)- b) ^ 2);
end

% call function to find nearest minutiae
XY = SelectNeighbours(length(x), angleXY, distanceXY);

% calculate weighted angular components for nearest minutiae
for k = 1 : length(XY)
          if(XY(k) ~= 0)
                    u = u +  cos(2 * theta(XY(k)))/distanceXY (XY(k));
                    v = v +  sin(2 * theta(XY(k)))/distanceXY (XY(k));
          end
end

m = 0.5 * (atan2(v, u));
%----------------------END OF FUNCTION--------------------------------

% finds the nearest minutaies
function[XY]= SelectNeighbours(size, angleXY, distanceXY)

xy = zeros([size 1]);

for k = 1: size
          xy(k) = k;
end

% arrange all minutiae by distance from block center
for k = 2: size
for j = 1: size - 1
          if(distanceXY(j) > distanceXY(j+1))
                    temp = angleXY(j);
                    angleXY(j) = angleXY(j+1);
                    angleXY(j+1) = temp;
                    temp = distanceXY(j);
                    distanceXY(j) = distanceXY(j+1);
                    distanceXY(j+1) = temp;
                    temp = xy(j);
                    xy(j) = xy(j+1);
                    xy(j+1) = temp;
          end
end
end

% traverse order distance array to find closest minutaie for sector by angle
```

```
sectors = 8;
XY = zeros([sectors 1]);
size_1 = zeros([sectors 1]);

for j = 1: sectors
for k = 1: size
        if(angleXY(k)>= (j-1)*2*pi/sectors && angleXY(k) < j*2*pi/sectors)
                if(size_1(j)~=1)
                        XY(j) = xy(k);
                        size_1 (j) = size_1(j) + 1;
                end
        end
end
end
%----------------------END OF FUNCTION-------------------------------
```

Appendix B.

```
/* header files */
#include <stdio.h>
#include <sys/types.h>
#include <dirent.h>
#include <errno.h>
#include <string.h>


/* maximum number of minutiae in image */
#define MAX 100

int main(int argc, char * argv [])
{

/* declaration of file pointers to read image and core */
FILE * fptr_image, * fptr_core;
/* declaration of file pointers to read write template shares */
FILE * fptr_share_1, * fptr_share_2;


/* declaration of directory and directory entry pointers */
DIR             *dip;
struct dirent   *dit;

/* declaration of arrays to store x, y, theta for each minutaie in file */
int x[MAX], y[MAX], theta[MAX];

/* string to store name of file */
char name[50];

/* to store the total number of minutaie for an image */
int size;

/* for core points */
int core_x  = 0;
int core_y  = 0;

/* loop counter */
int i;

/* check to see if user entered a directory name */
if (argc < 2)
{
                    printf("Usage: %s <directory>\n", argv[0]);
                    return 0;
}

/* open directory and validate where this operation was performed correctly */
if ((dip = opendir(argv[1])) == NULL)
{
                    perror("opendir");
                    return 0;
}


/* open core file */
if ((fptr_core = fopen(argv[2], "r")) == NULL)
{
                    perror("openfile: core");
                    return 0;
}

/* read from directory till all files have been processed */
while ((dit = readdir(dip)) != NULL)
{
          if(dit->d_name[0]=='.')
                    continue;
           size = 0;
           strcpy(name, argv[1]);
           strcat(name,"/");
           strcat(name, dit->d_name);
           /* open minutaie file */
           if ((fptr_image = fopen (name, "r")) == NULL)
           {
                     perror("openfile: image");

                     return 0;
           }

          /* read minutaie */
          while(!feof(fptr_image) && size < MAX )
          {

                    fscanf(fptr_image, "%d %d %d", & x[size],
                    & y[size], & theta[size]);
                    size = size + 1;
          }

          /* close minutaie file */
```

```
    fclose(fptr_image);


    /* read core */
    fscanf(fptr_core, "%s %d %d", &name, &core_x, &core_y);


    /* open template shares files to write to */
    strcpy(name, argv[3]);
    strcat(name,"/");
    strcat(name, dit->d_name);
    strcat(name, "_1");
    if ((fptr_share_1 = fopen(name, "w")) == NULL)
    {
        perror("openfile: shares_1");
        return 0;
    }

    strcpy(name, argv[3]);
    strcat(name,"/");
    strcat(name, dit->d_name);
    strcat(name, "_2");
    if ((fptr_share_2 = fopen(name, "w")) == NULL)
    {
        perror("openfile: shares_2");
        return 0;
    }

    i = 0;
    /* write minutaie in the two files based on the fingerprint template
splitting scheme */
    for (; i < size ; i ++)
    {
            if(y[i] < core_y)
             fprintf(fptr_share_1, "%d %d %d\n", x[i],
              y[i], theta[i]);

              else if(y[i] > core_y)
             fprintf(fptr_share_2, "%d %d %d\n", x[i],
             y[i], theta[i]);
    }

    /* close share files */
    fclose(fptr_share_1);
    fclose(fptr_share_2);
```

```
}

/* close core file */
fclose(fptr_core);

/* close directory */
closedir(dip);

return 0;
}
```