B.TECH. PROJECT REPORT

On

Privacy Protection of Biometric Templates

BY

Aravind Ravikumar (160001006)

Ashish Banala (160001014)

DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE

December 2019

Privacy Protection of Biometric Templates

A Project Report

Submitted in partial fulfillment of the requirements for the award of the degrees

of
BACHELOR OF TECHNOLOGY
in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:
Aravind Ravikumar (160001006)
Ashish Banala (160001014)

Guided by:
Dr.Surya Prakash
Associate Professor and HOD, Discipline of Computer Science and Engineering, IIT Indore

INDIAN INSTITUTE OF TECHNOLOGY INDORE
December 2019

CANDIDATE'S DECLARATION

We hereby declare that the project entitled **"Privacy Protection of Biometric Templates"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in **Computer Science and Engineering** completed under the supervision of **Dr.Surya Prakash, HOD, CSE, IIT Indore** is an authentic work.

Further, I/we declare that I/we have not submitted this work for the award of any other degree elsewhere.

Signature of Students with Date

CERTIFICATE by BTP Guide

This is to certify that the thesis entitled **"Privacy Protection of Biometric Templates"** and submitted by Aravind Ravikumar, Roll No. 160001006 and Ashish Banala, Roll No. 160001014, in partial fulfillment of the requirements for CS 493 B.Tech Project embodies the work done by them under my supervision. It is certified that the declaration made by the students is correct to the best of my knowledge.

Signature of Supervisor

Dr SURYA PRAKASH

Associate Professor

Discipline of Computer Science and Engineering

Indian Institute of Technology Indore

Acknowledgments

We would like to take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of our project.

We would like to express our sincere gratitude to **Dr. Surya Prakash**, our project supervisor, for his valuable suggestions and keen interest through out the progress of our project.

We would like to thank **Mr. Vivek Baghel** for his valuable suggestions throughout the project.

Last but not the least, we would like to thank our family members, classmates and other students of IIT Indore for their support.

With Sincere Regards,

Aravind Ravikumar

Ashish Banala

Abstract

Security and privacy of a biometric system plays a crucial role in it's functioning and acceptance. Leakage of biometric data can seriously undermine a person's privacy and security. Biometric cryptosystems eliminates this issue by storing in the database a secure sketch that is typically acquired by binding the template with a cryptographic key.

Biometric cryptosystems typically contain an algorithm to align the fingerprint images and another algorithm to generate the secure sketch from the aligned fingerprint image. As a part of this project we propose a novel fingerprint alignment technique based on Principal Component Analysis (PCA). We verified the correctness of above proposed technique by using it in conjunction with a modification of the Binarized Phase Spectrum (BiPS) representation of minutiae proposed in [1], which is a fixed-length binary string obtained by quantizing the Fourier phase spectrum of a minutia set.

We also propose a biometric cryptosystem comprising of features of triangles formed by the triplets of minutiae as the basic representation unit. This method will free from possible distortions in the fingerprint template like translation, rotation, scale, sheer, occlusion and clutter and consequently results in alignment free authentication. Subsequently we tested this technique by using it in conjunction with the BiPS representation proposed in [1].

Contents

List of Figures

List of Tables

Abbreviations

- BCS - Biometric Cryptosystem
- CB - Cancellable Biometrics
- PCA - Pricipal Component Analysis
- BiPS - Binarized Phase Spectrum
- OFFC - Orientation Field Flow Curve
- ICP - Iterative Closest Point
- BCH Code - Bose–Chaudhuri–Hocquenghem Code
- RS Code - Reed-Solomon Code
- RSC - Recursive Systematic Convolutional
- FRR - False Rejection Rate
- FAR - False Acceptance Rate

Chapter 1

Introduction

In the present automated world, human authentication has become a crucial component of all the systems and products around us. This increases the importance of automated authentication systems drastically. Computerisation of authentication system signifies that spurious authentication can lead to severe damages on many fronts such as data, privacy, finance etc.

Traditional automated authentication systems are based on tokens (ID cards or user name) and some knowledge (for example, password) known only by the user. Although these techniques have become highly popular,they have several hindrances and security concerns [8]. If an imposter gains access to the credentials of a user, traditional authentication systems fail to differentiate between the imposter and the genuine user. An authentication system based on biometrics alleviates such concerns.

Biometric systems function through computerised recognition of behavioural and physiological traits of an individual. These traits/characteristics can be properly obtained using appropriate sensors on the user. Biometric features are comparatively unique, permanent and cannot be shared and forged easily.

However biometric systems are vulnerable to a number of threats. Leakage of biometric template information can cause serious implications to the privacy and security of an individual. The imposter can reverse engineer the biometric traits of a person using the template data, thereby creating a spoof that can be used to gain authorisation. Since biometric traits of a user are irrevocable, leakage of biometric template information allows the imposter to gain access to various other

databases that the user has access to. Implications such as the ones mentioned above increase the significance of robust biometric template protection schemes.

1.1 Requirements of Template Protection Schemes

Some of the desired properties or requirements [8, 1, 3] of biometric template protection schemes are:

- **Security**: Demonstrates the irreversibility of any protection scheme. It should be computationally strenuous to reconstruct the original user biometric template from the stored reference data, hence preserving the users privacy.
- **Privacy**: Concerned with the unlinkability of the reference data to the user, especially by means of other biometric systems.
- **Performance**: On account of a template protection scheme, the verification performance of an authentication system must not degrade, especially in accuracy.
- **Revocability**: Possibility of generation of different versions of protected biometric templates based on the same biometric data (renewability). This becomes necessary in any scenario of compromise of data.
- **Diversity**: Protected templates, especially ones generated from the same user data should not have high degrees of resemblance of reference data features and should not allow cross-matching.

Even though several template protection schemes exist today, seldom we encounter a scheme that concurrently meets the above mentioned requirements, especially in handling a large variability in multiple acquisitions of identical biometric features, which often necessitates a tradeoff between the security and performance.

1.2 Classification of Template Protection Algorithms

There are several template protection schemes[2] existent today, and the major ones among these are:

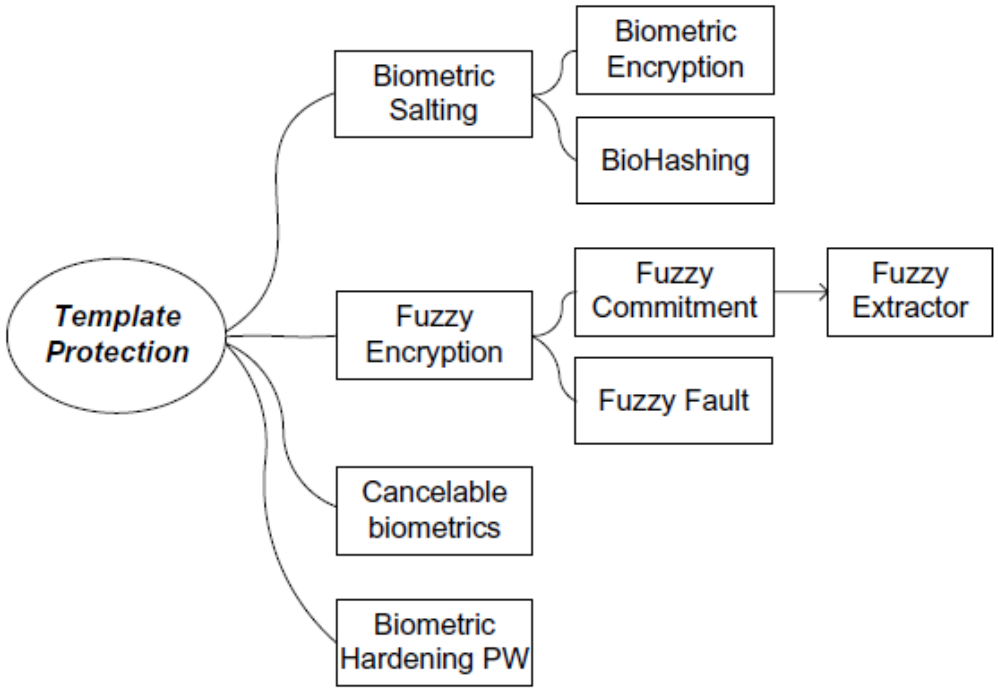

Figure 1.1: Types of Template Protection Algorithms [2]

- **Biometric Cryptosystem (Fuzzy Encryption)**: Biometric cryptosystems are designed to securely bind a digital key to a biometric or generate a digital key from a biometric [9, 3]. This combination of extracted biometric feature and the digital key is stored in the user reference database and is referred to as helper-data, hence this scheme is also referred to as helper-data based scheme. We will be using this template protection scheme for our biometric verification system and hence this will be outlined extensively in subsequent chapters.

- **Cancellable Biometrics (Feature Transformation)**: Cancellable biometrics consist of intentional, repeatable distortions of biometric signals

based on transforms which provide a comparison of biometric templates in the transformed domain [10]. The application of these transforms provides security(irreversibility) and privacy(unlinkability) to the user biometric templates. Evidently, cancellable biometrics are closely related to biometric cryptosystems.

One of the widely used approach for these transformations are non-invertible functions. In these approaches, biometric data are transformed applying a non-invertible function on the features extracted from the input biometric template. In order to provide updatable templates, parameters of the applied transforms are modified [3].

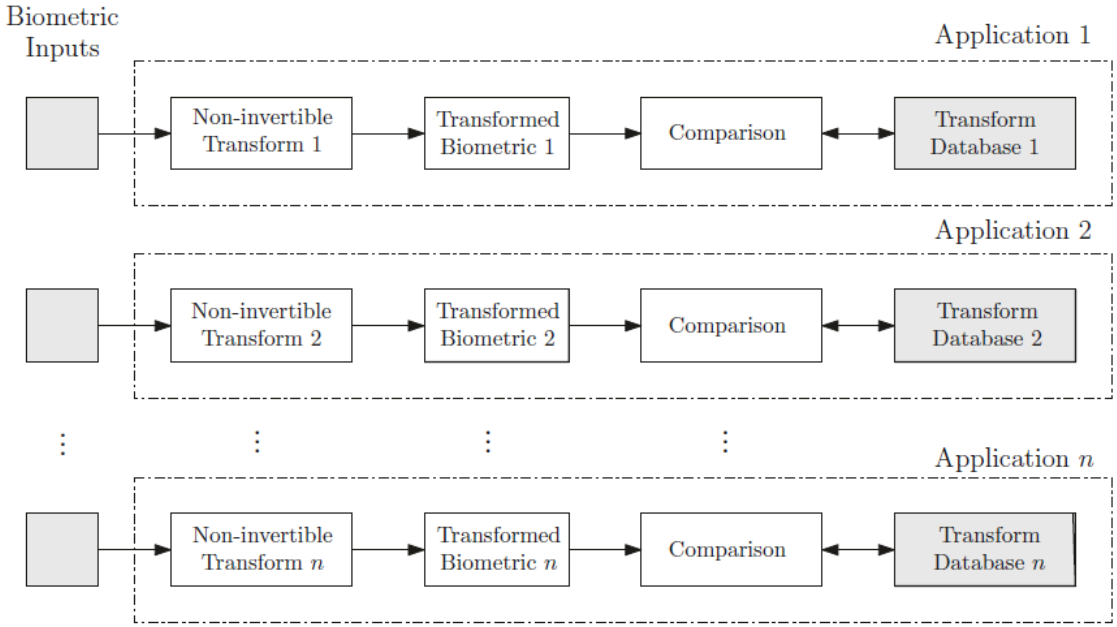

Figure 1.2: Cancellable Biometrics based on non-invertible transforms [3]

- **Biometric Salting**: Biometric salting functions by utilizing randomly generated patterns to distort the biometric template data. This method most often is sketched out as using a correlation filter to extract characteristic features from the biometric template and subsequently by multiplying both these characteristic features and moreover the filter with a predefined noise value, which contributes in providing a reference mask before embedding a randomly generated key in the reference using a lookup table [11]. Con-

sequently, the database will constitute of blurred correlation filter, lookup-table, and the hashed secret key [2].

1.3 Fingerprinting Nomenclature

Before we advance into finer details about biometric systems, these are some the nomenclature vital to the understanding of fingerprint biometrics and hence will be extensively used throughout this report.

- **Ridges**: The curved lines in a fingerprint template image is referred to as ridges. While most ridges are continuous curves originating and terminating at edges of the fingerprint image the others either terminate at specific points called ridge endings or perhaps a ridge separates into two to form specific points referred to as ridge bifurcation. These points are of significance, as location of these points correlates to uniqueness of fingerprint images.

  We first thin the ridges of any user biometric template to a single-pixel thick ridges so as to process the template with more ease and precision.

Figure 1.3: Original Template with Thinned-Ridge Template

- **Singular Point**: Singular point is defined as the topmost point of the innermost curving ridge. Noticeably, this will also be the point where the

corresponding ridge curvature reaches maximum in the whole template.

Figure 1.4: Singular Point depicted in a Fingerprint image

- **Minutiae Point**: Minutiae points are specific feature points in a fingerprint image. Minutiae points have characteristic that these are either locations of ridge bifurcation or of a ridge ending in the fingerprint.

  In Figure 1.5, red and orange dots depict points of ridge ending while blue and purple dots depict points of ridge bifurcation.

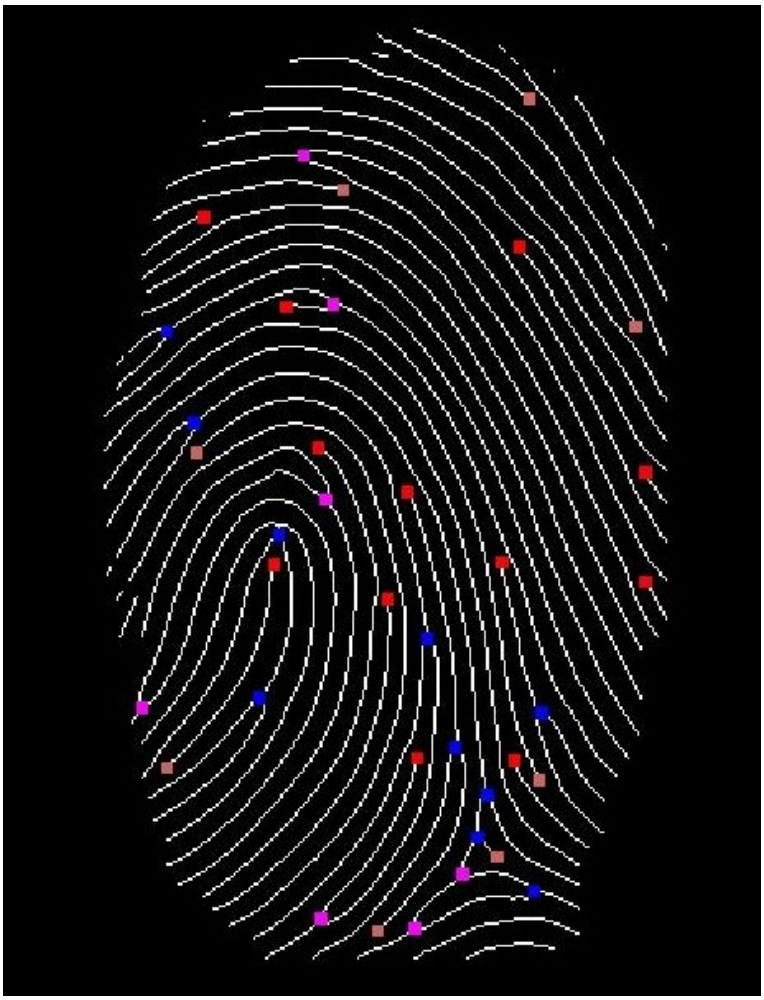

Figure 1.5: Minutiae Points on a Thinned and inverted fingerprint template.

Chapter 2

Literature Review

In this chapter, we'll look at the theories and concepts that have been used extensively in our project. This is in order to give a context to the implementation discussed in consequent chapters.

2.1 Biometric Cryptosystems

Biometric cryptosystems preserves information about the biometric features during the enrollment of the user template [12]. This feature data is represented as the helper data or the secure sketch, hence BCs are referred to also as helper data based schemes. This helper data assists in the course of verification to recover a cryptographically hashed key that was generated at the time of enrollement with the help of the preserved information (i.e. helper data) and the query biometric features. Authentication of the query template is carried out indirectly by verifying the validity of the extracted key generated from the fingerprint template and the secure sketch [1].

The verification of the hash of the key generaeted depicts an indirect authentication by performing comparison in the encrypted domain, which implies that biometric cryptosystems are applied as a means of a scheme for template protection [13]. Based on how the secure sketch is generated, biometric cryptosystems are classified [3] as either key-binding or key-generation systems:

- **Key-binding Schemes**: In this particular scheme, the secure sketch (helper data) is fabricated by the binding of a randomly generated key to the features

extracted from a given biometric template. As a result of this process of binding, we are left with a fusion of the randomly generated secret key and the biometric template is stored in the database and plays the role of helper-data in the reference information database. At the time of verification, the key is recovered using appropriate algorithms with the help of helper-data and the extracted features from the query biometric template [14]. The validity of this extracted key will consequently result in successful matching of the query template.

- **Key-generation Schemes**: In this particular scheme, helper-data is generated from the fingerprint template itself. The required keys are directly generated from the helper data and the biometric template [13]. Helper data-based key-generation schemes are also referred to as "fuzzy extractors" or "secure sketches" [12, 15].

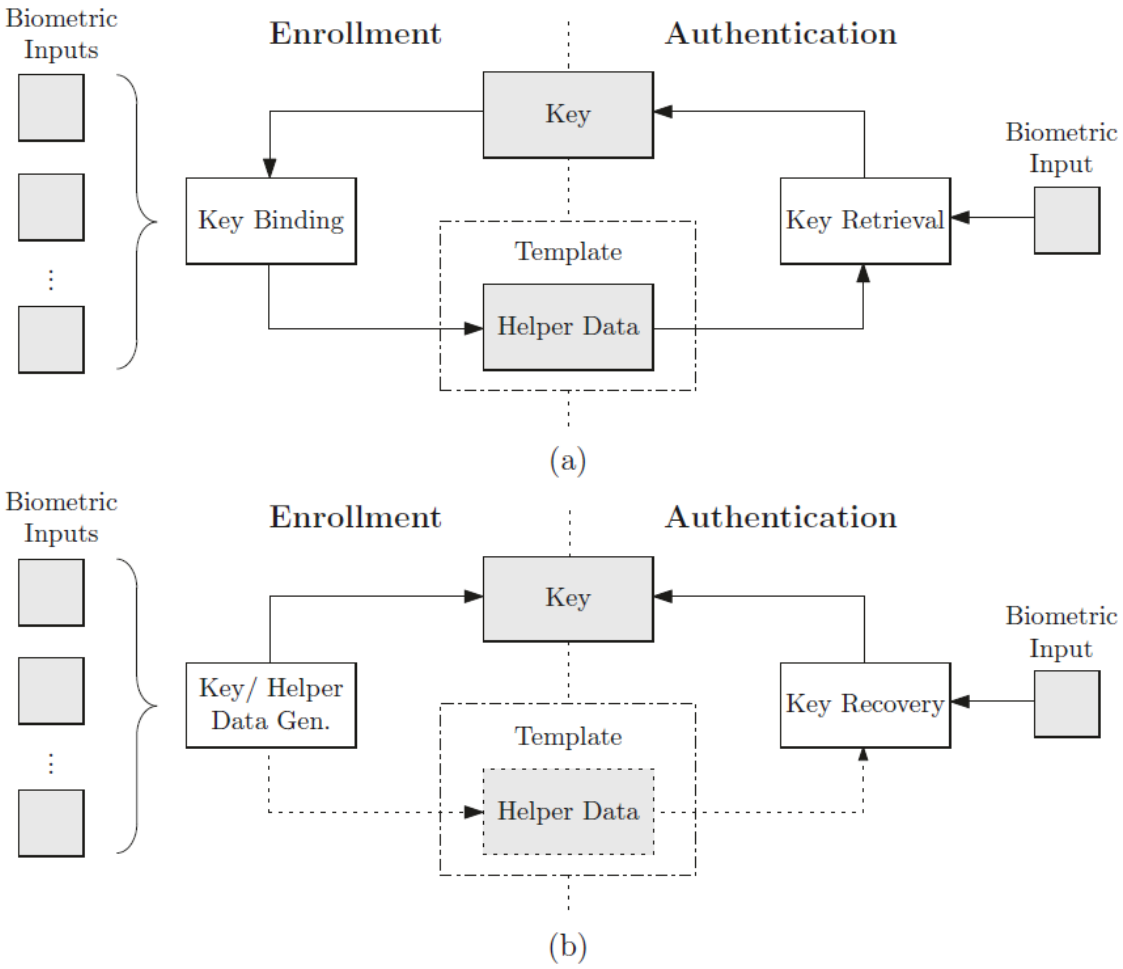

Figure 2.1: Basic Concept of (a)Key-Binding and (b)Key-Generation schemes [3]

For our study, we have chosen a key-binding scheme. As such, there are multiple approaches to implement biometric key-binding cryptosystems [3], lets go over the most widely used two:

- **Fuzzy Vault Scheme**: First introduced by Juels and Sudan [16] in 2006. The basic idea behind this scheme is the utilization of a hash set to lock a randomly generated secret key, resulting in a vault. Authentication is performed by matching with the constructed vault.

- **Fuzzy Commitment Scheme**: Introduced in 1999 by Juels and Wattenberg [17]. This method involves combination of techniques from the field of coding theory, error correction codes and cryptography to achieve a type of cryptographic primitive referred to as fuzzy commitment scheme. The authentication of a query fingerprint template depends on the similarity of the

generated binary string with the helper-data, verified using suitable error correction codes.

2.2 Orientation Field Flow Curves

The orientation field of a fingerprint template gives the direction of the ridge flow in a local neighborhood around all the points(sites) for all the ridges. The value of the orientation field at any site say s, will be o_s which is a vector $(\cos\theta_s, \sin\theta_s)^T$ where θ_s is the angle of the flow of the ridge with respect to the horizontal axis of the template image [18]. Opposite flow directions are equivalent, and thus the value of θ_s can uniquely be determined in the domain $(-\pi/2, \pi/2)$.

Figure 2.2: Orientation Field of a fingerprint template

Referencing [18], consider a site s in a fingerprint image I with r rows and c columns. The orientation field flow curve of I can be generated iteratively by

starting at a site starting point say $s_0 \in I$ and can be defined as

$$s_j = s_{j-1} + d_j.l_j.o_{s_{j-1}} \tag{2.1}$$

for $j = 1, 2, ...n$, d_j with value of either $-1, +1$ will be the flow direction from s_{j-1} to s_j, l_j is the length of the line segment from s_{j-1} to s_j and $o_{s_{j-1}}$ is the orientation field vector at site s_{j-1}. Termination occurs when either s_n reaches the boundary of the template or when iteration count exceeds a predefined value N_0. We will do this iteration for several s_0 chosen at the halfway point of the template and for both directions of d_j. Hence,

$$s_0 = (r_{start} + kw, c_{start} + lw), \tag{2.2}$$

with $k = [(r_{end} - r_{start})/2w]$ and $l = 1, 2, ...[(c_{end} - c_{start})/w]$. This implies that the starting points will be sampled across the vertical line that bisects the fingerprint template.

Figure 2.3: Orientation Field Flow Curve of a fingerprint template

Alignment of a fingerprint template using orientation field flow curves [19] is carried out by matching clusters of high curvature points of each of the orientation field flow curve using iterative closest point matching [1].

2.3 Error Correction Codes

Error correction codes were first designed in the field of information and coding theory [20]. The message that needs to be communicated is encoded by utilizing a suitable error correction code, which means that the message is being converted into a code word, by adding redundant bits of information. The code word is then sent through the possibly noisy channel and the received message is decoded by the

receiver utilizing the same error correction code to retrieve an output, resembling as closely as possible to the original message [4]. The degree of resemblance will depend on the bit-error rate of transmission and the error correction capability of the error correction code [21, 22].

The two primary types of error correction codes are Block Codes and Continuous(Convolutional) Codes.

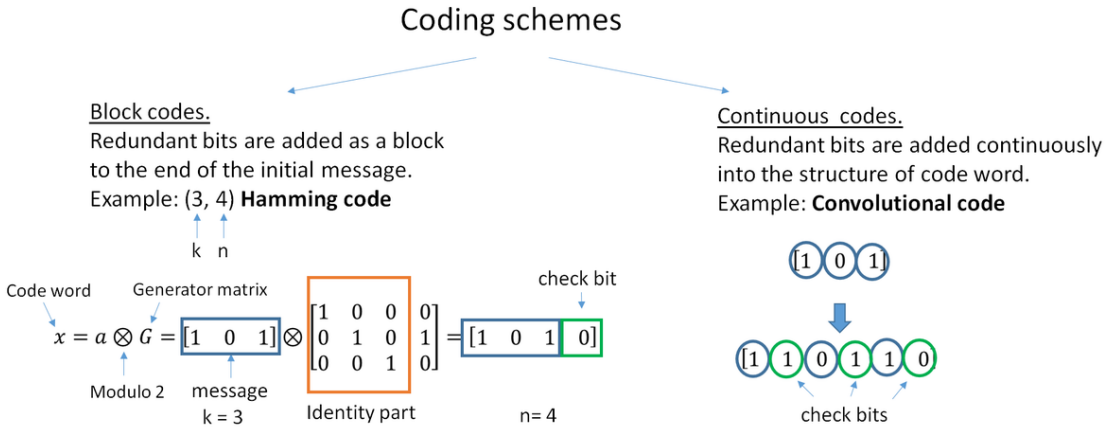

Figure 2.4: Classification of Error Correction Codes [4]

2.3.1 Block Codes

Block codes operate on blocks or packets having a fixed predetermined length of bits (predetrmined size). Generally, all block codes are generally hard-decoded in polynomial time respective to the corresponding length of blocks [23]. Most linear block codes are constructed making use of polynomials through a finite field, also widely known as Galois field.

Galois field arithmetic is widely implemented for the high speed error control coding applications based on the finite field $GF(p^m)$ with m usually being a small natural number and p being a prime of sufficient value to generate the required field size. Most of the device which performs function such as error control encoding, error detection and error correction operates by performing Galois field over $GF(p^m)$, due to the performance factor [22]. Major error correction codes like BCH codes and Reed-Solomon codes makes use of Galois field arithmetic.

BCH codes or Bose–Chaudhuri–Hocquenghem codes form a class of linear and cyclic error-correcting codes that are fromulated using polynomials over a finite field (Galois field) [24, 25]. One of the major advantage of using BCH codes is that during the design of code, we can control and predict exactly the error correction capability of the code. In particular, it is possible to design binary BCH codes that can correct multiple bit errors [26].

Reed-Solomon codes form a class of non-binary, linear, cyclic block codes which are basically q^m-ary versions of BCH code of length n dividing $q^m - 1$ that satisfy the Singleton bound [21, 27]. Consqeutively, this implies that RS code generalizes the binary versioned BCH codes over the Galois field $GF(q)$ [26].

2.3.2 Convolutional Codes

Convolutional code constitutes the other major classification of error correction codes. Convolutionary codes involve error-correcting codes that produce parity symbols by sliding a boolean polynomial function to the data needed for encoding [26]. The sliding application reflects the encoder's ' convolution ' over the data, resulting in the term ' convolutionary encoding ' [23, 28].

The state structure of the convolutional code which is being used is described by a structure known as the trellis structure. Trellis structure clearly defines the initial evolution and the convolution present in the codes [26].

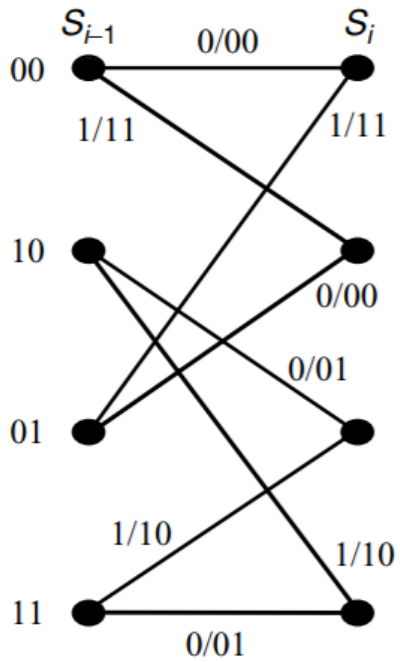

Figure 2.5: Trellis Structure of an RSC Encoder [5]

Turbo codes form a part of high performance codes which are known as forward error correction (FEC) codes. They were one of the first error correcting codes that allowed us to reach the theoretical maximum of transfer rate which is known as the Shannon limit. Turbo codes generally consist of a parallel concatenation of two convolutional codes. They are then decoded using iterative decoding algorithms [26].

In a turbo encoder's construction, Recursive systematic convolutional (RSC) encoders are used as the encoders E1 and E2 which have a rate of $R_c = 1/2$, such that $c_1' = c_1, c_2' = c_2$ and the lengths of the sequences m, c_1' and c_2', and c_1 and c_2 are all the same. In that case, the turbo encoding rate throughout is $R_c = 1/3$ [5].

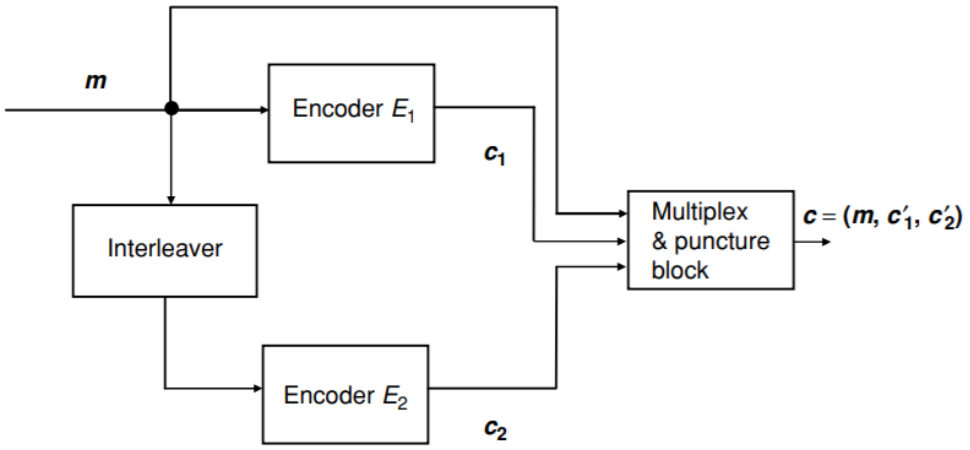

Figure 2.6: Turbo Code Encoder [5]

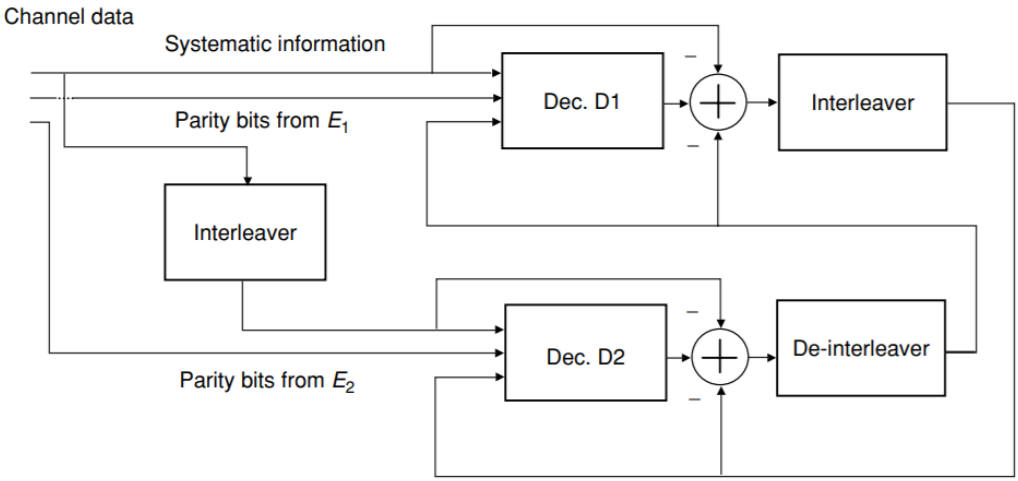

Figure 2.7: Turbo Code Decoder [5]

2.3.3 Application in Biometric Cryptosystems

The requirement of error correction codes in our project involves in using the error correction capability of codes to rectify minor differences between the binary string generated from the enrollment template and that of the query template.

Let, x^E be the binary string generated from the enrollment fingerprint template image after Fourier transforms of the extracted minutiae points afterwards representing it as a binarized phase spectrum and subsequently selecting the most

reliable N_c bits [1]. Hence, our enrollment template x^E is a binary string of length N_c.

Now we independently generate a random key k of length N_k bits. By encoding this key k by use of a suitable error correction code, we generate a code-word c of length N_c, which is same as that of our enrollement template. This is done by adding error correction bits to our key of length N_k to form the code-word of length N_c.

The secure sketch (helper data) stored in the database comprises of z and $Q(k)$, where $z = (x^E \oplus c)$ and $Q(.)$ is a cryptographic hash function. This helper data will be used at time of authentication to retrieve the generated key back with the help of the query template and the validity of this retrieved key will determine whether the query is matching or not.

Consequently, at time of authentication, authentication template x^A is generated from the query fingerprint image which is a binary string of identical length N_c. Now we extract the code-word c' (with errors) by utilizing authentication template x^A and helper data z by performing, $c' = x^A \oplus z = x^A \oplus (x^E \oplus c)$. This extracted code-word will be similar to the code-word generated at time of enrollment c, with the errors between c' and c being the errors between enrollment and authentication templates x^E and x^A respectively.

Now we decode c' by using the respective error correction code to obtain c^*. Consequently, if the difference between the authentication template x^A and enrollment template x^E is less than the error correction capability of our code, then the extracted code-word c^* will be identical to our initial code-word c. From c^*, we extract key k^* and successful match occurs when $Q(k^*) = Q(k)$, where $Q(k)$ is obtained from the helper data.

2.4 Pairing Function

A pairing function is utilized to uniquely generate a single natural number for every pair of natural numbers given as input [29]. By extending this definition generally, pairing functions can be used to encode a function defined on a set of natural numbers $f : N^k \Rightarrow N$ into a new function $g : N \Rightarrow N$[30].

The Cantor pairing function is a bijection [29] $\langle ., . \rangle : N^2 \Rightarrow N$ and is defined as

$$\langle x_1, x_2 \rangle = \frac{(x_1 + x_2)^2 + 3x_1 + x_2}{2}, \qquad \forall (x_1, x_2) \in N^2 \tag{2.3}$$

The core idea behind the pairing function is the counting of diagonals $x + y = k$ by increasing ordinates.

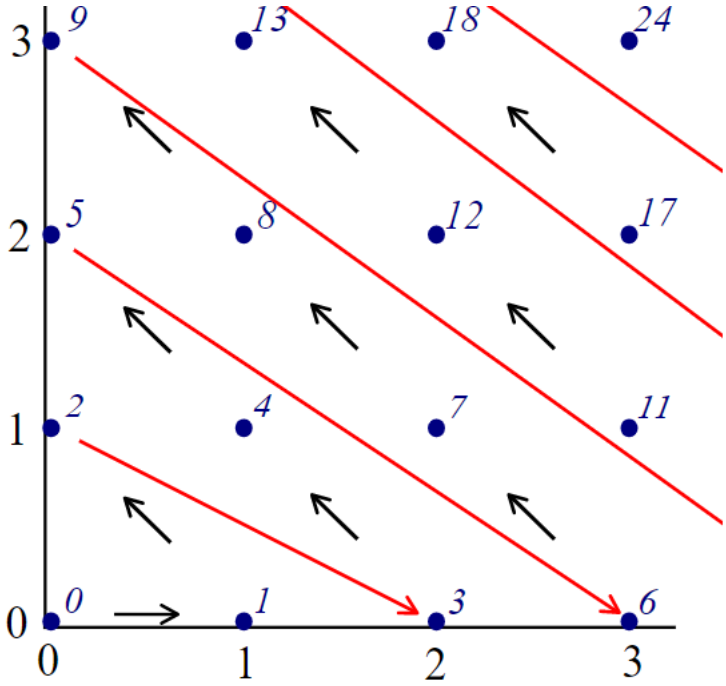

Figure 2.8: The Cantor Pairing Function [6]

Chapter 3

Proposed Methods

As a part of this project we propose a novel fingerprint alignment technique using Principal Component Analysis (PCA). In addition to this we propose a modification to the Binarized Phase Spectrum (BiPS) representation proposed in [1]. We have created a biometric cryptosystem using PCA as the alignment technique and the modified BiPS as the binary string generation technique and tested the working of this system.

We also propose a biometric cryptosystem comprising of features of triangles formed by the triplets of minutiae as the basic representation unit. This method will free from possible distortions in the fingerprint template like translation, rotation, scale, sheer, occlusion and clutter and consequently results in alignment free authentication.

3.1 Alignment Using PCA

Fingerprint alignment techniques form an integral part of many biometric template protection schemes. Some examples of fingerprint alignment techniques are, alignment based on focal point [1], Orientation Field Flow Curves [18].

Here we propose a novel fingerprint alignment technique based on Principal Component Analysis (PCA). PCA generates two orthogonal eigenvectors for a fingerprint image. These eigenvectors correspond to the axes along which the sum of projections of all the ridge points of the fingerprint is maximum

Figure 3.1: Fingerprint image with two orthogonal eigenvectors

Alignment of the fingerprint image is subsequently achieved by rotating the image so as to align the aforementioned axes along the x and y axes respectively. However performing this said rotation leads to a few issues. To fit the rotated image in frame, it might be the case that the dimensions of the fingerprint image have to be altered so as to not alter the size of ridges.

Considering the various possible orientations and sizes that the fingerprint image of a single finger can have, it can lead to stark differences in the dimensions of the enrolled fingerprint and the query fingerprint. Mathematically, let (x_p, y_p) be the location of a minutia point P in the enrolled fingerprint of a particular finger. Let a query fingerprint image corresponding to the same finger but a different orientation is presented, owing to the said rotation being performed, the location of the minutia point P can be some $(x_p^{'}, y_p^{'})$ which is completely different from the location of P in the enrolled fingerprint. This compromises the biometric system leading to high false rejection rates (frr).

Figure 3.2: Fingerprint image before and after performing the rotation to align eigenvectors along x and y axes

To avoid the above said problem which arises due to rotation, we propose the following method:

After the required rotation is performed, we extract the singular point whose location is (x_s, y_s) from the fingerprint image. Let's say that we were also able to extract n minutia points from the fingerprint image whose locations are denoted by (x_i, y_i) $i = 1, 2, .., n$ n is the number of minutiae points extracted from the fingerprint. Then we modify the locations of all minutiae points as follows,

$$(x_i - x_s, y_i - y_s) \; \forall i = 1, 2.., n \tag{3.1}$$

In the above operation we are essentially shifting the origin of the image to the singular point. Now we are only considering the locations of minutiae points relative to the singular point, which will be the same irrespective of the size of the fingerprint image. We are thereby eliminating the possibility of anomalies arising from rotation affecting the authentication process. This also improves the accuracy of the process.

3.2 Binarized Phase Spectrum (BiPS)

Fuzzy commitment is one of the approaches used in designing a fingerprint cryptosystem. As a part of fuzzy commitment, the minutia set extracted from a fingerprint image needs to be transformed into a fixed length binary string, which is then secured using fuzzy commitment.

Many methods have been previously proposed to achieve the above mentioned transformation. Some of the methods have been proposed by Nagar et al. [31]. Chang and Roy [32] and Xu et al. [33] ,

However the most promising , which was the *spectral minutiae representation*. Subsequently a modification to this in the Binarized Phase Spectrum (BiPS) was proposed in [1]. The spectral minutiae representation in [33] is obtained treating the minutia set as a collection of 2D Dirac-Delta functions and its Fourier transform is obtained after low pass filtering. The magnitude spectrum thus obtained in invariant to translation and rotation and hence facilitates alignment free matching.

The spectral minutiae representation however has a few shortcomings. Therefore [1] proposed BiPS representation. Spectral minutiae representation made use of the magnitude spectrum of the Fourier transform but ignored the phase spectrum. [1] proposed an exact opposite strategy where only the phase spectrum is considered and the magnitude spectrum is ignored.

The phase spectrum in turn can be sampled along a polar logarithmic grid to obtain the required minutiae representation and this representation can be quantized into two bits for each sample signifying the quadrant in which the phase lies. This method however requires an algorithm to align the fingerprint images using the features of the fingerprint.

3.2.1 Proposed Modification to BiPS

As a part of this project we propose a modification to the BiPS representation based on experimental results, which gave better results compared to BiPS [1] when used as a part of a biometric cryptosystem with PCA as the fingerprint alignment technique.

Let $\boldsymbol{M} = \{m_i\}$, $i = 1$ *to* n denote the set of all minutiae points, where n is the number of minutiae. Each minutia point m_i constitutes it's location (x_i, y_i) and direction θ_i. In [1] we associate a function $g(x, y)$ for each minutia point m_i as

$$g(x, y; m_i) = \delta(x - x_i, y - y_i)exp(j\theta_i) \tag{3.2}$$

Here the function δ represents the 2 Dimensional dirac delta function.

However, when PCA is used as an alignment algorithm better results have been achieved when the direction of the minutia point, θ_i is not considered. Therefore we propose a modified function for $g(x, y)$ for each minutia point m_i.

$$g(x, y; m_i) = \delta(x - x_i, y - y_i) \tag{3.3}$$

The 2-D function $f(x, y)$ that defines the minutiae set $\mathbf{M}$ and the continuous Fourier transform is,

$$f(x, y) = \sum_{i=1}^{n} \delta(x - x_i, y - y_i). \tag{3.4}$$

$$F(u, v) = \sum_{i=1}^{n} exp(j(2\pi(ux_i + vy_i))) \tag{3.5}$$

Let (u_j, v_j) denote the j^{th} frequency sample, $j = 1, .., N$. On a polar logarithmic grid $u = \nu\cos(\phi)$ and $v = \nu\sin(\phi)$, where ϕ is the radial angle and ν is the radial distance. We can then select N_ν logarithmically spaced samples between ν_{min} to ν_{max} and N_ϕ linearly spaced samples from 0 *to* π. Therefore we can have $N = N_\nu N_\phi$ samples in total.

Let $\Psi(F(u, v))$ denote the phase of the Fourier spectrum of f(x,y),

$$\Psi(F(u, v)) = arctan\left(\frac{\sum_{i=1}^{n}\sin(2\pi(ux_i + vy_i))}{\sum_{i=1}^{n}\cos(2\pi(ux_i + vy_i))}\right) \tag{3.6}$$

To binarize the above phase spectrum, we note that the value $\Psi(F(u, v))$ lies between 0 *and* 2π. That is the phase lies in one of the four quadrants. So the phase of each sample can be represented using 2 bits. Therefore we can generate a binary string $\mathbf{x} = [b_1, b_2, b_3, ..., b_{2N}]$ from the phase spectrum, where

$$b_{2j-1} = sgn(Re(F(u_j, v_j))) \tag{3.7}$$

$$b_{2j} = sgn(Im(F(u_j, v_j))) \tag{3.8}$$

and

$$sgn(x) = 1,\ if\ x >= 0,\ or$$

$$sgn(x) = 0,\ if\ x < 0$$

3.2.2 Securing the Modified BiPS

The modified BiPS representation thus generated is then secured using fuzzy commitment[17]. This is done in the following way:

Let's say the enrollment template $\mathbf{x}^E$ is a binary string consisting of N_c bits. A uniformly random key κ of length N_k bits is generated independent of the enrollment template. Error correcting bits are then added to κ in such a way to generate a codeword $\mathbf{c}$ of exactly the same length as $\mathbf{x}^E$ i.e, N_c. The error correcting bits are added using a turbo encoder with a rate of 1/4.

The secure sketch corresponding to a fingerprint consists of two components, one is the hash of the uniformly random key $\mathbf{Q}(\kappa)$, where $\mathbf{Q}$ is a hash function and the other is $\mathbf{z} = (\mathbf{x}^E \oplus \mathbf{c})$, where $\oplus$ represents the bit-wise xor operation.

During authentication the template $\mathbf{x}^A$ is generated from the query fingerprint presented. Here we use the data stored in the secure sketch to calculate $\mathbf{c}^{'}$

$$\mathbf{c}^{'} = \mathbf{x}^A \oplus \mathbf{z} = \mathbf{x}^A \oplus (\mathbf{x}^E \oplus \mathbf{c}) \tag{3.9}$$

Here $\mathbf{c}^{'}$ is codeword with errors. It is then decoded using a turbo decoder to give a key $\kappa^{'}$. The fingerprint is then authenticated if,

$$\mathbf{Q}(\kappa^{'}) = \mathbf{Q}(\kappa) \tag{3.10}$$

However to generate the enrollment template $\mathbf{x}^E$ of length N_c bits we need to select N_c most reliable bits from the modified BiPS representation. The bit-mask

corresponding to the bits selected is also stored as a part of the secure sketch. During enrollment the N_c bits represented by the bit-mask are selected from the modified BiPS generated from the query fingerprint image. These selected bits form $\mathbf{x}^A$. This necessitates the presence of an algorithm to find the reliability of bits in the modified BiPS.

3.2.3 Algorithm to Calculate the Reliability of Bits

An algorithm to find the reliability of the bits in BiPS was proposed in [1]. We will be using the same algorithm to find the reliability of bits in the proposed modified BiPS representation. The reliability R of the k^{th} bit is given by,

$$R(b_k) = 1 - \epsilon(\nu_j) exp\left(-\frac{|y|}{\sigma(\nu_j)}\right) \tag{3.11}$$

Here $y = Re(F(u_i, v_i)\ or\ Im(F(u_i, v_i))$ depending on whether it is an even bit or an odd bit and (u_i, v_i) is the sample used in the calculation of b_k. ν_j is the radial distance corresponding to the sample used. And

$$\epsilon(\nu_j) = k_1 + \left(\frac{\log \nu_{max} - \log \nu_j}{\log \nu_{max} - \log \nu_{min}}\right)(0.5 - k_1) \tag{3.12}$$

$$\sigma(\nu_j) = exp(k_2(\nu_j)^{k_3}) \tag{3.13}$$

Here $k_1,\ k_2\ and\ k_3$ are constant parameters that have to be learned through training data. Additionally we have the constraints,

$$0 < k_1 < 0.5$$

$$k_2 > 0\ and\ k_3 > 0$$

3.3 Feature Set from Minutiae Triplets

Different from the above mentioned methods, we developed a novel way based on utilizing local triangle feature set to match the possibly distorted fingerprint templates [7]. This method operates on extracting features from the triangles

formed by sets of minutiae triplet and making use of pairs of these features to generate binary string for the biometric template. The similarity between the triangle feature set is employed to characterize the similarity between fingerprint templates.

Figure 3.3: Local triangle sets formed in slightly distorted fingerprint templates

Efficient fingerprint recognition remains a difficult problem as the scale of the fingerprint image archive can be huge and major differences can exist between specific fingerprints. These distortions could include translations, rotations, scaling, shear changes, occlusions, and cluttering due to scars, dryness, sweat, smudge, etc [34]. An indexing algorithm based on innovative features formed by the minutiae triplets and are demonstrated by using triplets as the basic representation unit in a principled manner.

For each minute in a triplet, a simple and accurate descriptor can be defined. The following benefits are observed for algorithms based on minute triplets, which render them of potential higher quality than algorithms based on other representations [35]:

- These are tolerative of fingerprint deformations and can thus results in alignment free matching.

- These are suitable for inter-operability on standard based applications because the most common specifications are oriented on minutiae vetrices only.
- Minutiae triplets have greater discriminative strength than pairs of minutiae and single minutia representations, consequently accounting for stricter matching.

3.3.1 Triplet Based Features for Indexing

The list of local triangle (minutiae triplet) features that we utilize should be carefully chosen because of uncertainties of minutiae locations, that is, the location of each minutiae vertex changes independently in a small local area in a random manner [34]. Hence, the features that are possible prospects for generating binary string are the ones that show least variation to these minute coordinate changes. The features that are considered here are:

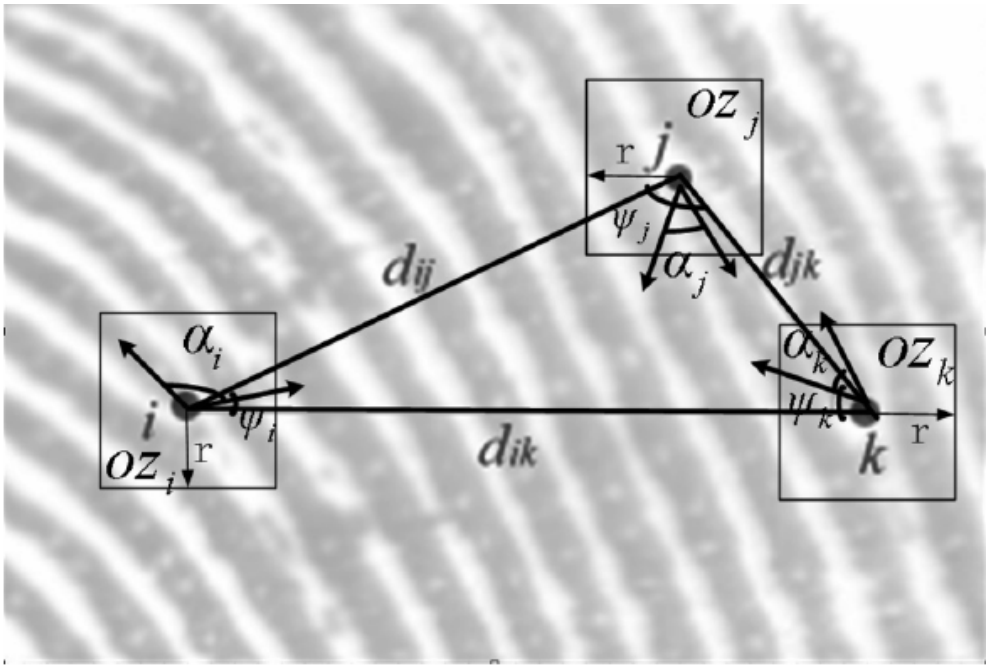

Figure 3.4: Local Triangle Features [7]

- **Angles**: Angles in the triangle formed by the minutiae triplets can be used for indexing. The minimum angle (α_{min}) and the median angle (α_{med}) show the least percentage change with respect to small changes in the minutiae

location [34]. Consequently, we consider only the minimum and median angles of the triangle as possible feature for indexing.

- **Length of sides**: Length of the sides of the triangle formed by triplets can be considered, especially the maximum length side as it constitutes a relatively unique characteristic among various sets of local triangle sets. Denoted as d_{ij}.

- **Orientation differences within the minutiae region**: Maximum orientation difference between the minutiae vertices within the minutiae region of the triangle forms another possible feature characteristic [7]. Denoted as OZ_i. For the square region surrounding minutiae i having coordinates (x_0, y_0) with a empirical parameter (distance from minutiae to side of the square) r, OZ_i is calculated as follows:

$$OZ_i = \frac{\sum_{i=x_0-r}^{i=x_0+r} \sum_{j=y_0-r}^{j=y_0+r} |Ori(i,j) - Ori(x_0, y_0)|}{2r * 2r} \tag{3.14}$$

  where $Ori(i, j)$ is the orientation at point (i, j) in the template, referred to as r in 3.4.

- **Orientation of the Minutiae vertices**: Various properties of the orientation of the minutiae vertices can be utilized as characteristic feature for local triangle sets. These can include angles formed by the orientation at a minutiae vertex with both of its sides and the angle formed by the orientation with the angle bisector of the triangle at the minutiae vertex, and the latter is denoted as ψ_i in 3.4.

- **Triangle handedness**: Minutiae are ordered on the basis of distance from the singular point of the template and handedness quantitates the order of minutiae vertices forming the triangle. Say, Z_{ij} is the vector from minutiae i to j. The handedness of the triangle will be $\phi = sign(Z_{ij} \times Z_{jk})$, and consequently ϕ will either be 1 or -1.

- **Area**: Even though variations in area are relatively larger than the other characteristic features, it settles in the pool of prospective charateristic fea-

tures. The area of a triangle can be calculated as,

$$A = \sqrt{s(s - d_{ij})(s - d_{jk})(s - d_{kl}} \tag{3.15}$$

where s is the semi-perimeter of the triangle and defined as, $s = \frac{d_{ij}+d_{jk}+d_{ki}}{2}$.

- **Triangle type**: This operates by generating a triangle type based on type of minutiae point, that is, if it is a ridge endpoint or point of ridge bifurcation. $\gamma_i = 1$ if it constitutes an endpoint otherwise 0 for ridge bifurcation. Triangle type of the local triangle is calculated as $\gamma = 4\gamma_i + 2\gamma_j + \gamma_k$. Hence, $0 \leq \gamma \geq 7$.

Chapter 4

Implementation and Results

The implementation and testing for both the proposed methods in the project has been done using MATLAB® and it's associated tool boxes. The following sections explain the details of the implementation, experimentation and the results.

4.1 Modified BiPS with PCA

The testing for the proposed biometric cryptosystem which uses PCA as the alignment technique and modified BiPS as the binary string representation of the fingerprint has been done using the following parameter settings: $\nu_{min} = 0.01,\ \nu_{max} = 0.25,\ N_\nu = 128,\ N_\phi = 37,\ k_1 = 0.46,\ k_2 = 10\ and\ k_3 = 0.4$.

In addition to these, the length of the codeword N_c has been set to 2047 bits and the length of the secret key N_k has been set to 67 bits. SHA-256 has been used as the hash function $\mathbf{Q}(\kappa)$ to generate the hash of the secret key. The error correcting capacity of the turbo decoder used during authentication is 550 bits. During the experimentation however, we have not taken all of the minutiae points into consideration to generate the proposed spectrum.

The testing was done using taking only $'n'$ minutiae points closest to the singular point. Since we have shifted the origin to the singular point, this means we have taken $'n'$ minutiae points with the smallest value of $\sqrt{x_i^2 + y_i^2}$, where (x_i, y_i) is the location of the i^th minutiae with singular point as the origin. Let m be the number of minutiae points extracted from the fingerprint. Hence the number of minutiae points taken into consideration $= min(n, m)$

Through experimentation, the optimal value of n was determined to be 14 which was the smallest value of n for which we were able to get zero False Acceptance Rate (FAR). Values of n less than 14 yielded lower False Rejection Rates (FRR) and non-zero FAR, owing to the lack of enough data to distinguish between fingerprints. Values of n greater than 15 yielded zero FAR but higher FRR values.

All the testing has been done on FVC2002-DB1 database. The testing was initially done using the BiPS representation proposed in [1] along with PCA as the alignment technique using the FVC protocol. Subsequently the testing has been done on the proposed modified BiPS with PCA using FVC protocol, and then using the 1v1 protocol. The aforementioned tests were done by enrolling a single fingerprint during enrollment. We also tested our proposed method by enrolling two fingerprint images and tested the same using FVC protocol, which gave significantly better results. Below is a tabulation of the results at zero FAR:

Algorithm and Protocol	**FRR**(%)
BiPS and PCA (FVC Protocol)	29.2
Modified BiPS and PCA (FVC)	17.8
Modified BiPS and PCA (*Enrolling two images)	9.1
Modified BiPS and PCA (1v1)	4

Table 4.1: FRR values of proposed biometric cryptosystems using various protocols at zero FAR

* The method followed during testing by enrolling two images is as follows: Each finger in the FVC2002-DB1 database has 8 samples. During enrollment, two samples of the same finger are enrolled. Then all the samples are authenticated using these two samples as the enrollment template. If a query fingerprint matches with either of the two enrolled samples, it is treated as a match. We enroll images in the following pairs and authenticate them against the remaining samples of the finger, $(1\_1, 1\_2), (1\_2, 1\_3), (1\_3, 1\_4)...(1\_8, 1\_1)$

For calculating FAR values, we enroll two samples of the same finger and

use a sample from each of the remaining fingers as a query fingerprint. A false acceptance occurs if the query image gets authenticated by atleast one of the enrolled images.

4.2 Minutiae Triplet Feature Extraction

Implementation is carried out by first extracting the minutiae points from the fingerprint template. Again, the minutiae set closest to the singular point of the fingerprint are chosen as to avoid edge case minutiae points that may not occur in another template of identical user. This will assist in avoiding the characterization of triangles that occur only accompanying templates that comprises of relatively extreme edge cases. Lets denote the set of these minutiae points as M, where M_i comprises of the coordinates (x_i, y_i) of the minutia point i, and the orientation of the ridge at the minutia point with respect to the horizontal axis, denoted as r_i.

We choose triplets of minutiae to form triangle feature set by filtering through the entire possible set of all triplets. This is done to filter down the total triangle set to only those that doesn't contribute as extremely distorted or obscured (for example, triangles having triplet of minutiae points linear, triangles having near-coinciding minutiae vertices or triangles spanning across the entire fingerprint template). For our project, we consider triplets whose generated triangle has the properties satisfying maximum length side less than 100 pixels and the maximum angle less than 120°. Hence, $max(d_{ij}, d_{jk}, d_{ki}) < 100$ and $max(\alpha_i, \alpha_j, \alpha_k) < 120°$.

Calculation of the features of the triangle generated by triplets can be carried out by passing parameters that are the coordinates (x_i, y_i) and the orientation with respect to the horizontal axis r of minutia vertex i. Lengths of sides of the triangle is computed by $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2)}$, angles of the triangle are calculated as $\alpha_i = \tan^{-1}(\frac{m_1 - m_2}{1 + m_1 m_2})$, where m_1 and m_2 are the slopes of the sides of the triangle having minutiae i as a vertex, angle between the orientation of the minutiae point and the sides of the triangle is also calculated in a similar manner by replacing m_1 by r_i and the area of the triangle is computed as $A = \sqrt{s(s - d_{ij})(s - d_{jk})(s - d_{kl}}$, where s is the semi-perimeter of the triangle and defined as, $s = \frac{d_{ij} + d_{jk} + d_{ki}}{2}$.

Now we normalize these characteristic features before utilizing them in generation of binary string. Normalization is required to prevent features of different measurements like distances and angles interacting with each other, which could result in one feature dominating over the others because of the lack of scale in between them. Hence, we normalize the distances with the length of the template image, which typically equals 300 pixels and the angles with the maximum angle that can be achieved one way, which equals 180°.

To incorporate more feature set characteristics, we use pairing functions to generate a unique natural number from two or more selected parameters, applying Cantor pairing function as defined in 2.4. This implies combining unique parameters like combining distance, angle pairs and using this as feature set for generating binary string.

Initially, we have to determine the features that we utilize to generate the binary string. For determining this, we will first divide our database into two sets, one will be our training set used to determine the best suited parameters to continue with and the other being the testing set which will be used as the performance measurement and the final results will be defined on this set.

The following results have been obtained by performing the verification on FVC2002 DB1-A database, which was the database used for the Second International Competition for Fingerprint Verification Algorithms. This database describes 8 fingerprint templates each for a domain of 100 users in total. We will use 3 templates per user for our initial test to determine the most suitable parameters for further testing.

We have also only considered set of minutaie points n (Here, $n = 15$) closest to the singular point of the template. We have followed identical FVC protocol, which operates by enrolling each template and authenticating with another template of identical user to calculate the False Rejection Rate (FRR) and by enrolling a template and authenticating a single template of all the other users to compute False Acceptance Rate (FAR).

Features Used	**FRR**(%)
Max Distance, Max Angle, Max of angle between orientation and sides of the same vertex	36.59
Max Distance, Max Angle, Area	34.29
Max Distance, Median Distance	37.10
Max Distance, Max of angle between orientation and sides of the same vertex	27.61
Max Distance, Median Distance, Min Distance	43.12
Max Distance, Area	47.32
Max Distance, Max Angle, Median Distance, Median Angle	36.82

Table 4.2: FRR Rates of Different Feature Sets of Minutiae Triplets

From the above results, we conclude that using maximum distance and the maximum of angle between orientation and sides of the same vertex as characteristic features to generate binary string from. This is majorly because the significance of a distance feature as a distinction between similar triangles of different scale and an angle feature that exhibits relatively minute variations to slight change in location of minutiae points due to possible distortions. Extensive results using these features are shown in 4.3.

Features Used	**FRR**(%)
FVC Protocol	29.11
FVC Protocol	20.54
1v1 Protocol (random)	16.39

Table 4.3: FRR Rates for the Best Performing Set of Features

Chapter 5

Conclusion and Future Work

As a part of this project we have proposed PCA as a novel alignment technique. We have also proposed a modification to BiPS, which can be used as the binary string generation technique in a biometric cryptosystem.

We have created a biometric cryptosystem which uses PCA as the alignment technique and the modified BiPS as the binary string generation technique. By testing this biometric cryptosystem using the FVC2002 database we have shown that the said system gives better results when compared with the biometric cryptosystem which uses PCA and BiPS proposed in [1]. Thus we have verified the usage of PCA as an alignment technique in biometric cryptosystems and also that our proposed modification to BiPS gives better results especially when used in conjunction with PCA. The modification to BiPS representation also reduces the computational complexity incurred during enrollment and authentication by not taking into consideration the orientation of individual minutiae points.

Furthermore, as part of this project we have also proposed a novel technique to utilize features of triangles formed by the triplets of minutiae as the basic representation unit and consequently use these triangle characteristic features for generation of binary string using modified BiPS. This technique is unaffected from possible distortions in the fingerprint template and therefore results in alignment free verification. With further work on better extraction of characteristic features from triangle sets, this method will provide significant advantages over single and double minutiae feature methods.

Bibliography

[1] K. Nandakumar. A fingerprint cryptosystem based on minutiae phase spectrum. In *2010 IEEE International Workshop on Information Forensics and Security*, pages 1–6, Dec 2010.

[2] Christoph Busch Arjan Kuijper Xuebing Zhou, Stephen D. Wolthusen. A security analysis of biometric template protection schemes. Springer, Berlin, Heidelberg, December 2009.

[3] Christian Rathgeb and Andreas Uhl. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, 2011(1):3, Sep 2011.

[4] Wikipedia: Error correction codes, Dec 2019.

[5] J. Castiñeira Moreira and P. G. Farrell. *Turbo Codes*, chapter 7, pages 209–275. John Wiley and Sons, Ltd, 2006.

[6] Wikipedia: Pairing function, Dec 2019.

[7] Xinjian Chen, Jie Tian, Xin Yang, and Yangyang Zhang. An algorithm for distorted fingerprint matching based on local triangle feature set. *IEEE Transactions on Information Forensics and Security*, 1(2):169–177, June 2006.

[8] S. S. Ali, I. I. Ganapathi, and S. Prakash. Robust technique for fingerprint template protection. *IET Biometrics*, 7(6):536–549, 2018.

[9] Ann Cavoukian and Alex Stoianov. Biometric encryption chapter from the encyclopedia of biometrics. 2009.

[10] N. K. Ratha, J. H. Connell, and R. M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001.

[11] Nalini K. Ratha ; Sharat Chikkerur ; Jonathan H. Connell ; Ruud M. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, February 2007.

[12] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan L. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, pages 523–540, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[13] Anil Jain, Karthik Nandakumar, and Abhishek Nagar. Biometric template security. *EURASIP Journal on Advances in Signal Processing*, 2008, 03 2008.

[14] U. Uludag, S. Pankanti, S. Prabhakar, and A. K. Jain. Biometric cryptosystems: issues and challenges. *Proceedings of the IEEE*, 92(6):948–960, June 2004.

[15] E. A. Verbitskiy, P. Tuyls, C. Obi, B. Schoenmakers, and B. Skoric. Key extraction from general nondiscrete signals. *IEEE Transactions on Information Forensics and Security*, 5(2):269–279, June 2010.

[16] Ari Juels and Madhu Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38(2):237–257, Feb 2006.

[17] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. *Proceedings of the ACM Conference on Computer and Communications Security*, 1, 12 1999.

[18] Sarat Dass and Anil Jain. Fingerprint classification using orientation field flow curves. pages 650–655, 01 2004.

[19] K. Nandakumar, A. K. Jain, and S. Pankanti. Fingerprint-based fuzzy vault: Implementation and performance. *IEEE Transactions on Information Forensics and Security*, 2(4):744–757, Dec 2007.

[20] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[21] Jennifer Key. Some error-correcting codes and their applications. 05 2003.

[22] Mukesh Kestwal, Sumit Joshi, and Deepak Rawat. A review on error correcting codes deployed in mobility prone digital communications. *IJAIR, ISSN: 2278-7844*, 2:139–143, 01 2013.

[23] Bernard Sklar and F.J. Harris. The abcs of linear block codes. *Signal Processing Magazine, IEEE*, 21:14 – 35, 08 2004.

[24] Alexis Hocquenghem. Codes correcteurs d'erreurs. *Chiffres*, 2(2):147–56, 1959.

[25] R.C. Bose and D.K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and Control*, 3(1):68 – 79, 1960.

[26] Takahiro Yamada. 3 - block codes. In Hideki Imai, editor, *Essentials of Error-Control Coding Techniques*, pages 39 – 59. Academic Press, 1990.

[27] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[28] D. J. Costello, J. Hagenauer, H. Imai, and S. B. Wicker. Applications of error-control coding. *IEEE Transactions on Information Theory*, 44(6):2531–2560, Oct 1998.

[29] Lisi Meri. Some remarks on the cantor pairing function some remarks on the cantor pairing function. *Le Matematiche*, 62, 12 2007.

[30] Martin D. Davis, Ron Sigal, and Elaine J. Weyuker. *Computability, Complexity, and Languages (2Nd Ed.): Fundamentals of Theoretical Computer Science*. Academic Press Professional, Inc., San Diego, CA, USA, 1994.

[31] A. Nagar, S. Rane, and A. Vetro. Privacy and security of features extracted from minutiae aggregates. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1826–1829, March 2010.

[32] Ee-Chien Chang and Sujoy Roy. Robust extraction of secret bits from minutiae. In Seong-Whan Lee and Stan Z. Li, editors, *Advances in Biometrics*, pages 750–759, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[33] H. Xu, R. N. J. Veldhuis, A. M. Bazen, T. A. M. Kevenaar, T.A.H.M Akkermans, and B. Gokberk. Fingerprint verification using spectral minutiae representations. *IEEE Transactions on Information Forensics and Security*, 4(3):397–409, Sept 2009.

[34] Bir Bhanu and Xuejun Tan. Fingerprint indexing based on novel features of minutiae triplets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):616–622, May 2003.

[35] Miguel Angel Medina-Pérez, Milton García-Borroto, Andres Eduardo Gutierrez-Rodríguez, and Leopoldo Altamirano-Robles. Improving fingerprint verification using minutiae triplets. *Sensors*, 12(3):3418–3437, 2012.