Interoperable Protected Fingerprint Minutiae Templates

Xue Li

Masteroppgave
Master i Teknologi - Medieteknikk
30 ECTS
Avdeling for informatikk og medieteknikk
Høgskolen i Gjøvik, 2012

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

Interoperable Protected Fingerprint Minutiae Templates

Xue Li

2012/07/12

Abstract

With the increasing use of biometric enabled systems, a large amount of biometric systems, sensors, feature formats and matching algorithms have been developed and various template protection methods have been proposed in the literature to enhance the user privacy of biometric enabled systems. The issue of interoperability among different biometric systems has to be concerned. In this project, a fingerprint template protection method named RIME is proposed. It enables interoperability among fingerprint authentication/verification systems that use ISO minutiae template. The RIME method extracts features from minutiae triplets and transforms the features using random projection. In this thesis, the accuracy, irreversibility and unlinkability of RIME are tested and evaluated. The application of RIME under two factor authentication scenario is also discussed.

Keywords: Fingerprint identification, template protection, interoperability

Acknowledgements

I would like to thank my supervisor Christoph Busch for providing me suggestions and guidance on the project, helping me define the scope of my thesis, giving me suggestions on the thesis writing.

I would also like to thank the senior research Bian Yang for giving me guidance and source of information on the field of biometric template protection, and Guoqiang Li for providing me software support. It was a great time to work together with Bian Yang and Guoqiang Li in the laboratory at Gjøvik University College during this one semester work.

Finally I would like to thank my dear father and mother for the enormous support, help and love I have got from them, thank for their support not only during this work but also during the two years of master study.

Xue Li, 2012/07/12

Contents

List of Figures

List of Tables

1 Introduction

1.1 Topic covered by the project

A biometric system is the process of recognizing a person based on the person's physiological and/or behavioural traits[2]. Biometric authentication/verification services are more and more used in the society in the last decades. For example, EU visa information system will include Biometric technique to prevent people who are refused a visa by one Schengen country from applying to other Schengen countries, and India will implement a biometric identity program for its 1.2 billion population. With the increasing of using biometric systems, the user privacy issue has gained more attention from the public. For example, the UK stopped their work on storing biometric information in the passport since it was not accepted by the public because of the privacy issue. The leakage of stored biometric information can cause critical damage to the user privacy. For example, a adversary can figure out privacy concerned information (age, health condition, etc.) of the owner from the stolen template, link across the different databases by using the same biometric trait to get a profile of the victim's social activities, and even forge a fake biometric sample to get access to other systems. Unlike passwords or tokens, it is hard for a user to switch to a new biometric identifier once his/her biometric template is compromised, since biometric traits are limited, permanent and unique to an individual. Both the risk and cost are high for changing the biometric characteristics by some means such as surgery. Thus, template protection technologies have been developed in the recent years to prevent biometric data from leakage. Defined in [3], template protection is a approach to ensuring the security of the biometric templates while maintaining the recognition performance. It aims at preventing plain text biometric data from leakage. In a biometric authentication/verification system, biometric template protection is used on the biometric references stored in the database. As illustrated in Figure 1, by using biometric template protection, the plain text biometric template is converted into a protected template, then the comparison of a query and the stored references is done by using a pseudonymous identifier comparator in the protected domain instead of in the plain text domain.

This master thesis project focuses on minutiae template protection for fingerprint authentication/verification systems that employ ISO standard minutiae format[4]. The proposed method applies on ISO minutiae format compliant templates and enables performance interoperability among the fingerprint authentication/verification systems that use this format.

1.2 Problem description

Various template protection methods have been proposed in the literature to enhance the user privacy of biometric enabled systems. In the meanwhile, a large amount of biometric systems, sensors, feature formats and matching algorithms. have been developed. The issue of interoper-

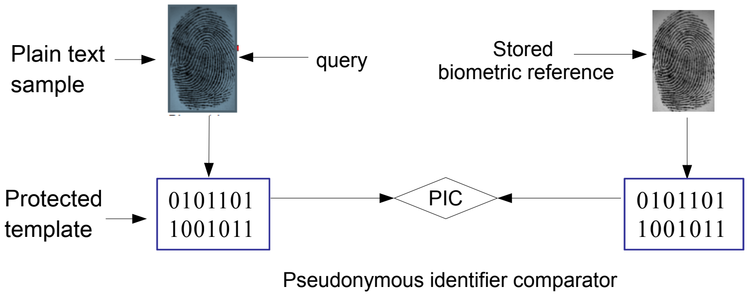


Figure 1: Biometric template protection concept

ability among different biometric systems has to be concerned.

Several standardization works [5][6] have been done to promote interoperability among biometric systems and made some progress. The interoperability on the plain text level can be realized by using standard template formats, but there is still a long way to go to achieve the interoperability on the protected template level. As stated in [7], "today, there is really very little interoperability among templates and the matching algorithms. Those are proprietary technologies". It is almost impossible to authenticate a biometric query which uses a specific template protection method if the reference data was generated by another method. The recent research on interoperability has not covered a complete, end-to-end, interoperable biometric system that employs techniques to protect the privacy of the subjects[8]. The biometric templates are not freely comparable across different systems, as illustrated in Figure 2. Due to the lack of interoperability among different biometric applications, customers have to pay switching costs if they want to switch from one service to another. The lack of interoperability also brings the risk of vendor lock-in, which means that customers have to depend on a single vendor for biometric services. For the biometric system developers, they need to be concerned that the vendor they are working with now will support their product in several years, since the cost of re-enrolling all the subjects could be significant [7].

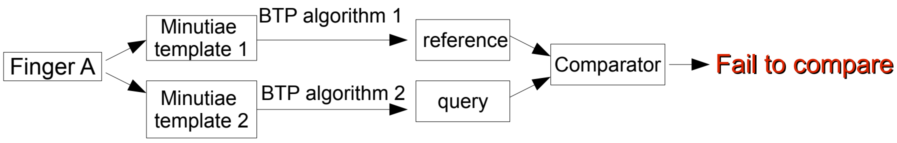


Figure 2: Interoperability problem

1.3 Justification, motivation and benefits

An interoperable scheme that can support all types of sensors, features and template formates and matching algorithms is difficult to realize[8]. As presented in [8], interoperability can be achieved by two steps: (1) Convert a biometric sample into a modality-dependent, predefined biometric feature data format; (2) Convert the modality-dependent, predefined biometric feature data to a protected template using a predefined format and method. For fingerprint authentication/verification systems, the first step can be achieved by using the existing standard formats, such as the ISO minutiae template format. In a ISO minutiae format template, a fingerprint sample is presented by a set of minutiae which is the ending or bifurcation of the ridges. Minutiae is presented by three dimensions, x, y and θ. x and y describe the location of the minutiae in a rectangular plane coordinate system, and θ is the ridge orientation at the minutiae, as illustrated in Figure 3.

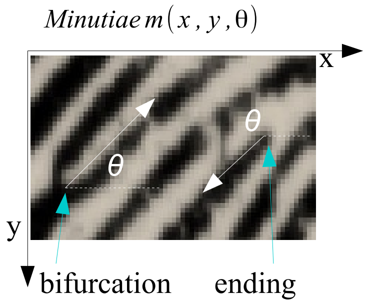


Figure 3: An example of ISO standard compliant minutiae

For the second step, if a template protection scheme which outputs standardized templates is designed, it will ensure that biometric subsystems from different providers can generate templates that meet the same format [9]. The minutiae template protection method proposed in this thesis achieves the second step. The proposed method takes ISO minutiae templates as input and outputs ISO minutiae standard compliant protected templates. Thus, the protected templates from different fingerprint authentication/verification systems that employs the proposed method are comparable by using a minutiae template comparator, as illustrated in Figure 4. The interoperability on the minutiae level is achieved.

The achievement of interoperability will reduce the dependency on a single supplier and the cost of switching templates format. Customers will have a better choice on which biometric system product to use. It will also reduce the risk of vendor lock-in and promote information sharing.

1.4 Thesis outline

Chapter 2 provides the basic knowledge of biometric systems and presents the related work in the filed of biometric template protection and the performance evaluation of biometric template protection methods.

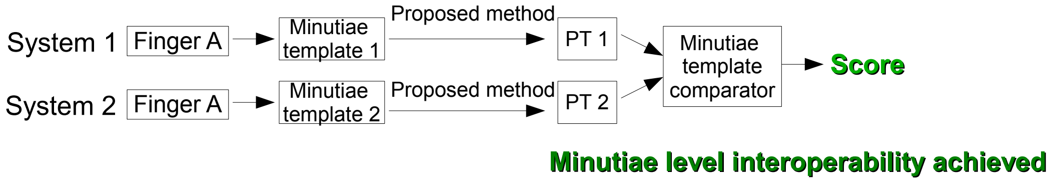


Figure 4: Achievement of interoperability on minutiae level by using the proposed template protection method.

Chapter 3 gives the methodology used in each step, including feature extraction, algorithm design and performance evaluation, of the project.

Chapter 4 presents the features that are extracted from minutiae templates and can be used in the proposed minutiae template protection method. The statistical characters, including probability density distribution, entropy and correlation, of the features are analyzed.

Chapter 5 gives the detailed algorithm of the proposed interoperable minutiae template protection method.

Chapter 6 presents the identification accuracy experiment results of the proposed method, and Chapter 7 presents the irreversibility and unlinkability assessment for the proposed method.

Chapter 8 discusses the application of the proposed method under two factor authentication scenario, and compares the accuracy of the proposed method under two factor authentication scenario with biometric template protection scenario.

Chapter 9 discusses the accuracy performance and interoperability of the proposed method. The influence of feature selection on the accuracy of the proposed method and the reason for performance degradation are discussed.

Chapter 10 summaries the contributions of this project and concludes the project.

Chapter 11 gives the possible improvements for the proposed method as feature work.

2 Related Work

This chapter presents the background knowledge for this master thesis project and the related work of biometric template protection. First, we introduce the basics of biometric systems. Then, we introduce the privacy issue of biometric systems and the concept of privacy enhancing techniques. Next, we have an overview of the existing biometric template protection methods. At the end, we introduce the criteria for evaluating the performance of biometric template protection methods.

2.1 Biometric systems

A biometric system is the process of recognizing a person based on a specific physiological and/or behavioural traits possessed by that person[2]. The traits includes fingerprint, face, iris, hand geometry, palmprint, vein, voice, gait, signature and DNA.

As shown in Figure 5, a generic biometric system has five major components[10]:

- A data capture subsystem: it is the interface between users and the system, which contains biometric capture devices or sensors that collect biometric characteristics from users and converts them into a biometric sample.
- A signal processing subsystem: it extracts the feature set which is useful in distinguishing different users from the biometric sample captured by the data capture subsystem.
- A data storage subsystem: it stores all the templates from the users and their identity references. The biometric feature extracted in the enrolment process is stored as the biometric reference and the identity the user claims in the enrolment process is stored as the identity reference for the identification and verification process.
- A comparison subsystem: it calculates how well the template from the sensor and the template stored in the database matches and outputs a comparison score. The higher the score is, the higher is the similarity between them.
- A decision subsystem: it make the decision depending on a threshold and the comparison score, and initiates a response to the query.

There are three functional processes employed in a biometric system, as shown in Figure 6:

- Enrolment process: during this process, the captured biometric sample is processed. A feature set is extracted from the sample and enrolled as a reference in the database with the identity reference. In the enrollment phase, the data from all the individual users is stored in the database. After the biometric sample is provided by the user, the system performs a quick quality control of the sample. Failure to enroll(FTE) is the percentage of times that users are not able to be enrolled in the system. FTE errors typically occur when the quality control fails.

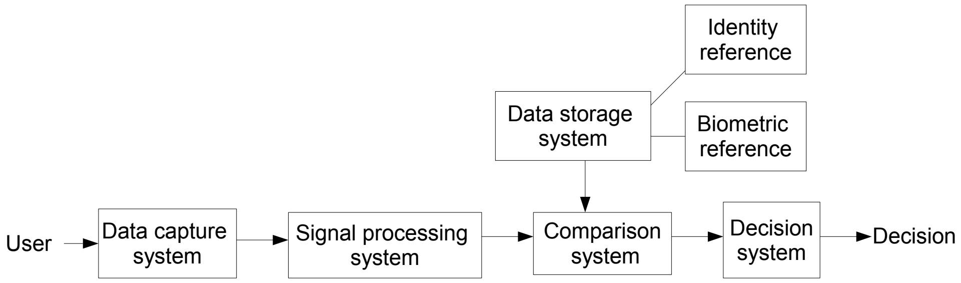


Figure 5: Simple diagram showing the main subsystems of a biometric system

This is to ensure that only reliable and usable biometric data is stored in the database[2].

- Identification process. The purpose of this process is to answer the question "Who I am?". It is a one-to-many mapping. During this process, the captured sample is compared against all references in the database and the list of individuals whose references match with the captured sample is returned.
- Verification process. The purpose of this process is to answer the question "Am I who I say I am?". It is a one-to-one mapping. It checks if an individual is the person that he/she claims to be. The user provides his/her biometric characteristic(s) to the capturing device and presents a claim of his/her identity. The captured sample is compared with the biometric reference linked to the identity reference for the claimed identity.

The standardized metrics for measuring the accuracy of biometric systems and biometric recognition algorithms are defined in[11]. False acceptance rate (FAR) and false rejection rate (FRR) are two main performance measures of biometric systems. During the verification phase, a genuine user could be falsely rejected and a non-valid user could be falsely accepted. This is known as false acceptance rate (FAR) and false rejection rate (FRR). The false acceptance rate (FAR) is a measure of the likelihood that the system falsely accepts an access attempt from an unauthorized user. The false rejection rate (FRR) is a measure of the likelihood that the system falsely rejects an access attempt from an authorized user.

False match rate(FMR) and false non-match rate (FMR) are two measures for the performance of the matching algorithms. A false match happens when the matching algorithm classifies an imposter probe as a genuine one, while a false non-match happens when the matching algorithm classifies a genuine probe as an imposter. The decision of a comparison depends on the comparison score and the chosen threshold. The FMR value for the threshold t is the proportion of impostors that get a comparison score higher than t among all impostor attempts. The FNMR value for the threshold t is the proportion of genuine comparisons with a comparison score lower than t among all genuine attempts. By choosing different thresholds, different FMR and FNMR can be obtain, see Figure 7.

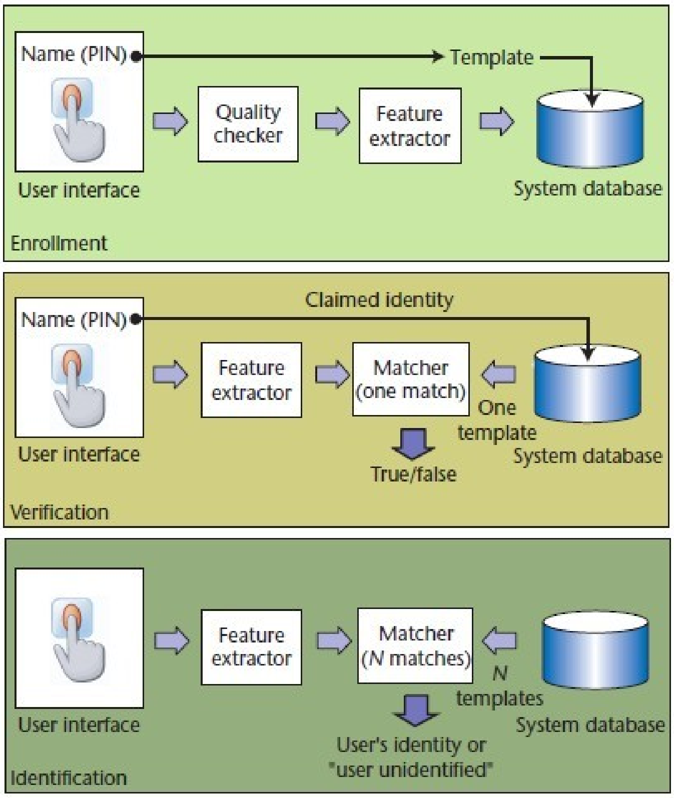


Figure 6: Block diagrams of enrollment, verification, and identification process, from[1]

By varying the threshold and plot FMR on the x-axis and FNMR on the y-axis, a DET (Detection error trade-off) curve is obtained. EER (the equal error rate) is the value where FMR and FNMR are equal, see Figure8.

2.2 Privacy and privacy enhancing techniques

When using a non-biometric system for authentication, for example passwords, to prevent compromising the user's password in the database, it is normal to hash the password instead of storing it in clear text. The most commonly used hash algorithms are md5 and sha-1. If a user shares his/her user name and password with a friend, the friend can access the resources that the user possesses. And there is no way to positively link the usage of the recourse to the actual user, so there is no protection against the repudiation of the user ID owner. Biometric technology can provide a much more accurate and reliable user authentication method[12], but raises other concerns:

1. Biometrics are not secret. Biometrics can be recorded and misused without the user's consent.
2. Biometric traits can not be revoked or cancelled since biometric traits, such as fingerprints and face, are limited, permanent and unique to an individual.

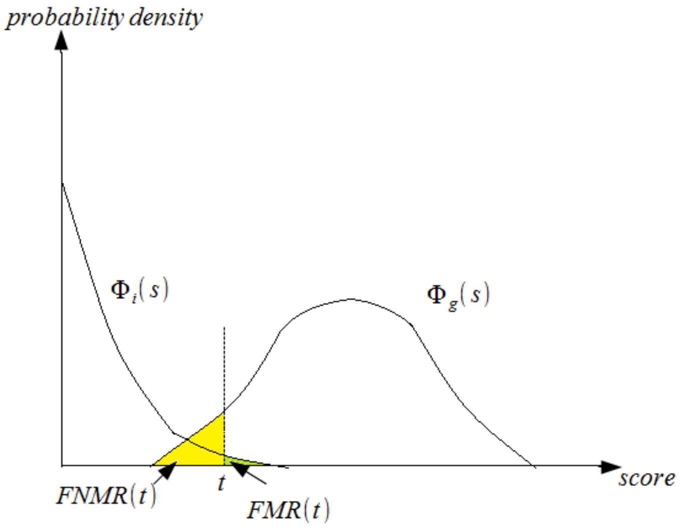


Figure 7: FMR and FNMR under the threshold t. $\Phi_i(s)$ and $\Phi_g(s)$ are the probability density function of the comparison score values from imposter attempts and genuine attempts respectively. t is the threshold. The yellow area is the FNMR and the green area is the FMR.

3. A compromised biometric is forever compromised. All the applications that use the biometric are compromised.
4. Cross-site matching can be used to track the users. If organizations share their databases, an attacker can link the information in different database together to get a social activity profile of the users.

As the increasing use and share of biometric data, privacy and security issues are increasingly concerned by the public. The leakage of stored biometric information can lead to:

1. Exposure of the user's sensitive information such, as health condition and age.
2. Cross matching of different databases, which mean one sample can be used to get access to several systems. This brings the risk of profiling attack, means that the attacker can link the information from different enrolled applications, such as bank records, financial records, health care records, to get a profile of the victim's social activities.
3. Faking biometric samples, means that the attacker can forge a fake biometric sample from the leaked biometric information to get access to other systems. This brings the risk of identity theft.

Thus, the user privacy needs to be enhanced to prevent illegal access to the applications and misuse of personal biometric information. This is important for biometric authentication/identification applications to gain acceptance and trust from the public.

When using biometric traits for authentication, it is hard to reproduce the exactly same data

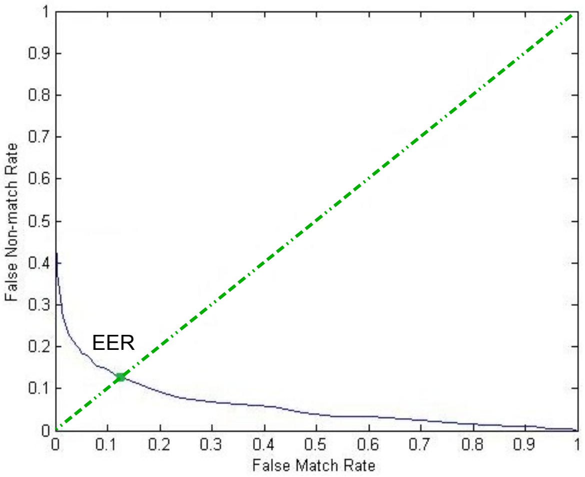


Figure 8: DET example

as the biometric data captured as reference. Thus, encryption algorithms such as hashing algorithms can not be used to protect biometric data because of the avalanche effect, "When even small changes of the input of a hash function result in a significant change of the hash values, the hash function possesses a strong avalanche effect"[13]. In order to protect the privacy, biometric template protection techniques have been invented. Using these techniques, pseudo identities [8] can be derived from biometric data to perform authentication instead of the original biometric samples. Thus, the original biometric information is protected against leakage.

2.3 Criteria of evaluating template protection methods

The criteria related to the protection properties of biometric template protection methods are defined and summarised in [14][15][3]. The criteria can be grouped into three categories: technical performance, operational performance and protection performance. The protection performance evaluation is mainly introduced in this section.

The operational performance[14] of biometric template protection methods includes modality independence, interoperability, variation of criteria and criteria dependencies. The definitions of these properties are explained in [14]. To evaluate the technical performance of a biometric template protection method, the following aspects need to be concerned:

1. Accuracy. The accuracy of the biometric identification algorithm is commonly measured by FMR, FNMR and EER.
2. Throughput, which is "the number of biometric transactions processed continuously by an individual biometric processing unit in a defined time interval".
3. Accuracy degradation. If we observe the identification accuracy over plain text templates and protected templates, accuracy degradation will occur after applying template protection

methods.

4. Diversity. It refers to the "maximum number of independent protected templates that can be generated from the same biometric feature by a biometric template protection algorithm".

To assess the protection performance of biometric template protection methods, security and privacy protection performances need to be evaluated. To provide effective security and privacy protection, a template protection scheme needs to fit the following requests:

1. Irreversibility. The biometric sample or features that used to generate a protected template should be transformed in such a way that the original biometric sample or features cannot be retrieved from the protected template, as illustrated in Figure 9. Irreversibility measures the difficulty of retrieving.
2. Unlinkability. Biometric references used in various applications should be unlinkable, meaning that a adversary cannot figure out whether two biometric references from two different applications are generated from the same biometric sample or not, as illustrated in Figure 10. Unlinkability measures the difficulty of classifying protected templates over time and accross applications[14].
3. Revocability and renewability. Revocability refers to that it should be possible to revoke a template and generate a new template from the same original data. Renewability refers to the ability to update a protected template. In [15], renewability is defined as a term that covers diversification capacity, irreversibility and unlinkability aspects. Revocability and renewability solve the issue of compromised references. The biometric traits are limited. Revocability and renewability ensures that various different references can be extracted from the same biometric sample, thus the references can be revoked or renewed once compromised.
4. Confidentiality and integrity. Confidentiality ensures that biometric data is not leaked to unauthorized entities. Integrity ensures that the accuracy and completeness of assets is protected.

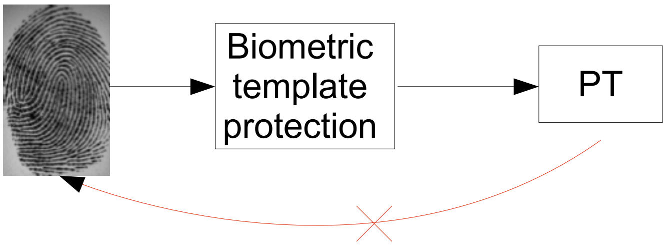


Figure 9: Irreversibility

In [15], security refers to the requirements on the system level and privacy refers to the requirements on the information level. Security includes the confidentiality and integrity of biometric data, renewability and revocability of biometric references. Confidentiality and integrity of biometric data and revocability can be achieved by system-level countermeasures, for example,

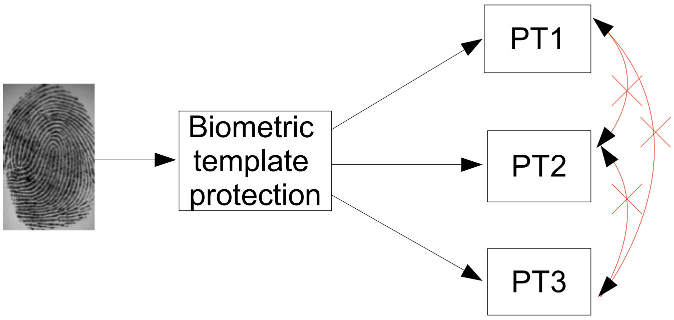


Figure 10: Unlinkability

the confidentiality can be achieved by access control and the revocability can be achieved by removing a compromised reference from the system. Therefore, they are not criteria for evaluating the protection performance of biometric template protection methods. Privacy refers to the irreversibility and unlinkability of protected templates. Irreversibility and unlinkability are important criteria for the protection performance o biometric template protection methods. They are also unique to biometric template protection methods.

2.4 Biometric template protection methods

A number of the approaches for securing the biometric templates have been proposed in the literature. In [16], biometric template protection schemes are classified into two categories: feature transformation and biometric cryptosystem. In the feature transform approach, the biometric template is transformed by using a function F and only the transformed template is stored in the database. The same transformation function is applied to queries and the comparison between references and queries is performed in the transformed domain. Feature transformation can be classified into two subcategories: biohashing and non-invertible transformation, depending on the transformation function is invertible or not. Biometric cryptosystems combine biometrics and cryptography to perform biometric matching in the cryptographic domain[17]. Some helper data which contains information from the biometric template and the encryption key is used. This helper data does not reveal much information about the key or the biometric template. Usually the helper data is an association of an error correcting code and the biometric template. It is used to extract a cryptographic key from the biometric query. A single entity that embeds both the key and the template is stored in the database as helper data. When a biometric query differs from the template within certain error tolerance, the helper data can recover the embedded key from the query. Recovery of the correct key implies a successful match[3].

2.4.1 Biohashing

Biohashing is also called salting. By applying a biohashing approach, biometric features are first extracted from the biometric sample and then transformed using a function defined by a user-specific key or password. In the feature transformation phase, the transformation function is invertible, means that if an attacker gets the key and the transformed template, he can recover the original biometric template. Hence, the security of Biohashing methods depends on the security of the key.

In a Biohashing scheme, it is very important to measure the entropy of a output protected template. In the biometric context, the entropy of a biometric template refers to the information content of the template. It measures how distinctive a template is. Low template entropy means that the templates generated by this Biohashing scheme are hard to be distinguished from one another. Thus, in a Biohashing scheme, a key should increase the entropy of the biometric template to make it difficult for the adversary to guess the template[16].

In [18][19] and [20], a biohashing framework called constituent random multispace quantization (RMQ) is used. There are three steps in RMQ:

1. Biometric data projection. In this step, the plain-text biometric data is projected into a lower-dimensioned and more discriminative feature domain. Two of widely used data projection methods are Linear Discriminant Analysis (LDA)[21] and Principal Component Analysis (PCA)[22]. PCA projects high dimensional data to a lower dimension to reduce data dimensionality. It keeps most of the sample's variation and is useful for the compression and classification of data. LDA performs dimensionality reduction and maximizes the ratio of between-class scatter to that of within-class scatter. Hence, it finds directions along which the classes are best separated and keeps as much the class discriminatory information as possible.
2. Random multispace mapping. In this step, the biometric feature set derived from the step 1 is projected onto a randomly selected set of orthogonal subspaces. The set of the subspaces is determined by an external input.
3. Quantization. In this step, a RMQ template is generated by quantizing the feature set from step 2 into a binary {0, 1}. The threshold for binarization is selected based on the criteria that the expected number of zeros in the template is equal to the expected number of ones so as to maximize the entropy of the template.

During authentication, the query RMQ template is compared with the reference template. The similarity is measured by Hamming distance.

2.4.2 Non-invertible transformation

Different from Biohashing methods, non-invertible transformation methods apply a one-way transformation function on the template. The parameters of the transformation function are defined by a user specific key. It is computationally hard to invert a transformed template even if the key is known.

The IBM proposed three non-invertible transformation methods in [23], namely, Cartesian, polar and functional transformation. They are also referred as cancelable biometrics. A cancelable biometric template protection approach is "an intentional, repeatable distortion of a biometric signal based on a chosen transform"[12]. They use a non-invertible transformation function that distorts the input biometric data before feature extraction or modifies the extracted feature set (e.g., minutiae points) itself. The protected template and the original template are formed by the same set of features. Templates that are formed by the features as the original feature set after the application of a non-invertible transform have been referred to as cancelable templates. For example, by applying cancelable fingerprints, the minutiae location and orientations are transformed irreversibly to generate protected templates such that a minutiae matcher can still be applied on the protected templates.

Cancelable biometric template protection methods consist of two steps: registration and transformation. We use cancelable fingerprint template protection as an example. In the registration step, all minutiae is pre-aligned with regard to some singular points. Core and delta can be chosen as a singular point, then other minutiae points can be aligned with regard to them. Once the global registration has been accomplished, the minutiae feature points can be transformed. The general idea of transformation is to irreversibly transform the minutiae positions and orientations.

In Cartesian transformation, the feature positions are measured in rectangular coordinates. The core is chosen as the reference. The x-axis is aligned by the core's orientation. This coordinate system is divided into cells of fixed size. The transformation is to change the cell positions. Multiple cells can be mapped to the same cell[24]. This means that the attacker can not guess the original template by analyzing the transformed template, this meets the requirement of irreversibility.

In polar transformation, the minutiae positions are measured in radial coordinates. The position and orientation of the core is used as a reference to align the positions and angles of other minutiae points. The coordinate space is divided into polar sectors. The transformation is to change the sector positions. The mapping is governed by a restricted translation key that defines the positions of sectors before and after transformation. The mapping is many-to-one as Cartesian transformation to ensure irreversibility. This transformation does not alter the original distribution of minutiae points[24]

In both Cartesian and polar transformation, a small change of minutiae position in the original fingerprint can lead to a large change in minutiae position after transformation if the minutiae point crosses a boundary of rectangles or sectors.

In functional transformation, a two-dimensional Gaussian function is used to move the minutiae points. The center and shape of Gaussian kernel is determined by the user-specific key. These Gaussian kernels overlap to form two surfaces. Then, they are used to decide the direction and amount of shift for each minutiae point. In order to transform a minutia, functions consisting of a mixture of Gaussians and its derivatives are evaluated at the position of minutia and then the minutia is translated according to the values obtained.

These three methods require alignment before transformation. The alignment accuracy depends on the accuracy of detecting the position and angle of singular points. In[23], these methods assume that core point can be detected accurately and all minutiae points are aligned with regard to the core before transformation. But practically, approximately 10% of automatic core-point extraction fails[25]. Several approaches to determine the core and delta has been developed, but it is still difficult to do precise estimation.

2.4.3 Fuzzy schemes

Fuzzy schemes utilize helper data and key binding cryptosystems to encrypt the biometric data. Two well known fuzzy schemes are fuzzy vault[26] and fuzzy commitment[27].

Fuzzy vault is an encryption scheme[26] that combines error correction and secret sharing. The principle of fuzzy vaults is:

> "Alice places a secret K in a vault and locks it using an unordered set A. Bob uses an unordered set B to unlock the vault(get access to K) successful if and only if B and A overlaps substantially"[26].

The procedure of fuzzy vault is:

> "Alice selects a polynomial p of variable x that encodes k, by fixing the coefficient of p according to k. Then Alice computes the polynomial projections, p(A), and adds some randomly generated points to it(chaff points), that do not lie on p, to arrive at a final point set R. When Bob tries to learn k(i.e. finding p), he uses his own unordered set B. If B overlaps with A substantially, he will be able to locate many points in R that lie on p. Using error-correcting coding, it is assumed that he can reconstruct p(and hence k). The security of this is based on the infeasibility of the polynomial reconstruction problem, so if Bob does not know about many points that lie on p, he can not easily find the parameter p. Without knowing p he can not access to k" [28].

A fingerprint template protection scheme using fuzzy vault is introduced in [28]. It consists two steps:

1. Enrollment: In the enrollment phase, the user provides his/her fingerprint. Then, some chaff points are added to the fingerprint template, this is then stored in a table called enrollment table. There exists information about each feature point, so when chaff points are added, they are placed outside the range where the real feature points are. If the chaff points and the feature points are located very close, it could lead to a mismatch for a valid user. The information for each of the chaff points is selected randomly. Then the hole fingerprints, consisting of real feature points and chaff points, are stored in the template database[28].
2. Verification: In the verification phase, the chaff points and the real feature points from the enrollment table should be separated. The feature points is extracted from the query of the fingerprint and then stored in a table called verification table. The enrollment table and the verification table are compared with each other. If the two tables overlap substantially, the key can be reconstructed.

The implementations of fuzzy vault for fingerprint [29] [30], face [31] and iris [32] have been proposed in the literature.

Fuzzy commitment takes biometric templates as private keys and employs error correcting codes (ECC) to solve the fuzziness problem of biometric templates[33]. It uses an error correcting code to account the difference between the reference and the query. The fuzzy commitment scheme consists two steps:

1. Committing, which is the enrollment phase. To commit a biometric data string x, we generate a codeword c from x according to a specified error correcting code. Then, we apply a hash function to c to get the commitment: $(\mathrm{hash}(c), \delta)$, where $\delta = x \oplus c$, $\mathrm{hash}(c)$ is the hash of the codeword c.
2. Decommitment, which is the verification phase. To decommit a commitment, the user provide a biometric query string x', we calculate $c' = x' \oplus \delta$ and $f(c')$, where f is the decoding function for the error correcting code. Then, we calculate $\mathrm{hash}(f(c'))$. $\mathrm{hash}(f(c)) = \mathrm{hash}(f(c'))$ if and only if $x \approx x'$ up to a certain error correction threshold. If $\mathrm{hash}(f(c)) = \mathrm{hash}(f(c'))$, the query is authenticated.

An implementation of fuzzy commitment for iris verification system is presented in [34].

2.4.4 Secure sketches and fuzzy extractors

Secure sketches and fuzzy extractors are key generating approaches. The idea behind these approaches is to extract a reliable cryptographic key from noisy biometric data. Some public information P is derived from the original biometric template to generate the key. This public information P is called a sketch[35]. An unformal definition of a secure sketch is provided in [36]: a secure sketch P is some information derived from noisy data X such that P does not reveal too much information about X, and given a Y that is similar to X according to some similarity measure so that X can be reconstructed from Y and P.

In a secure sketch approach, in the enrollment phase, it generates a sketch P from the original biometric template X . In the verification phase, for a query Y , according to the properties of secure sketches, if Y and X are similar according to some similarity measure, X can be reconstructed from Y and P, the reconstructed template $X' = X$.

The implementations of secure sketch for fingerprint [37] [38] face [39] template protection have been proposed.

3 Methodology

Statistical [40], mathematical[41] and quantitative [42] methods are used in this master thesis project. An interoperable minutiae template protection algorithm is designed, tested and evaluated using these methods.

First, we use statistical methods to analyse the statistical characters of the features derived from minutiae templates. The probability density distribution, entropy, dynamic range of the features and the correlation among the features are analyzed. By analyzing the statistical characters of the features, we find the minutiae-based features that are reliable to use as transformation parameters in the proposed template protection method.

Then, mathematical methods are used to extract features from the minutiae templates and transform the features. The proposed minutiae template protection method is a feature transformation method. The transformation function uses the reliable features as transformation parameters and projects the features onto a randomly selected orthogonal space. A number of parameters are used to adjust the feature values before the transformation and the dynamic range of the output after the transformation. The output protected templates are a set of three dimensional points. The values of the points and their ranges should be appropriate to ensure the matching performance.

Finally, quantitative methods will be used to estimate the accuracy performance, unlinkability and irreversibility of the proposed method. The accuracy of the propose method using different sets of features and parameter settings are tested. The irreversibility against brute force attacks is discussed. The unlinkability is analyzed from the aspects of similarity and distance between two templates.

4 Triplet-Based Features

We derive features from minutiae templates to use them as the transformation parameters in the proposed method. In this chapter, we present the feature extraction method and analyze the probability density distribution, entropy, dynamic range and correlation of the features.

4.1 Basic functions

In this subsection, we adopt some definitions from [43] to calculate the features used in the proposed template protection method. The functions are as follows:

In this thesis, minutiae is presented as $m(x, y, \theta)$ or $M(x, y, \theta)$, where x and y are the x-coordinate and y-coordinate of the minutiae in a rectangular coordinate system respectively, and θ is the ridge orientation of the minutiae. Given two minutiae $m_i(x_i, y_i, \theta_i)$ and $m_j(x_j, y_j, \theta_j)$, we have the following functions:

1. $\vec{v}_{ij}$, which is the vector with the initial point m_i and terminal point m_j:

$$\vec{v}_{ij} = (x_j - x_i, y_j - y_i) \tag{4.1}$$

2. $dir(\vec{v}_{ij})$, which is the direction of the vector $\vec{v}_{ij}$:

$$dir(\vec{v}_{ij}) = \begin{cases} arctan(\frac{\Delta y}{\Delta x}) & if \Delta x > 0 \wedge \Delta y \geq 0 \\ arctan(\frac{\Delta y}{\Delta x}) + 90° & if \Delta x > 0 \wedge \Delta y < 0 \\ arctan(\frac{\Delta y}{\Delta x}) + 180° & if \Delta x < 0 \\ 90° & if \Delta x = 0 \wedge \Delta y > 0 \\ 270° & if \Delta x = 0 \wedge \Delta y < 0 \end{cases} \tag{4.2}$$

where $\Delta x = x_i - x_j$, $\Delta y = y_i - y_j$, the results of $arctan(x)$ are limited to the interval $(-90°, 90°)$.

3. ed_{ij}, which is the length of the vector $\vec{v}_{ij}$, also the Euclidean distance between m_i and m_j:

$$ed_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{4.3}$$

4. $angle(\alpha, \beta)$, which is the angle required to rotate the vector with the angle β anticlockwise to superpose it to the vector with the same origin but the angle α, $\alpha \in [0, 360°)$, $\beta \in [0, 360°)$:

$$angle(\alpha, \beta) = \begin{cases} \alpha - \beta & if \alpha > \beta \\ \alpha - \beta + 360° & otherwise \end{cases} \tag{4.4}$$

5. $angle_{within 180°}(\alpha, \beta)$, which is the angle that is less than or equal to 180° formed by the vector with the angle α and the vector with the angle β:

$$angle_{within 180°}(\alpha, \beta) = min(\mid \alpha - \beta \mid, 360° - \mid \alpha - \beta \mid) \tag{4.5}$$

4.2 Triplet-based feature representation

In this section, we present the features that are extracted from minutiae templates and used in the proposed template protection algorithm. We form triplets from the minutia templates and derive features from the triplets.

Given a minutiae template MT_f from finger f and a radius value r. MT_f contains N_{MT_f} minutiae. For the i^{th} minutiae M_i in MT_f, we first form a circle $c(i,r)$ which has M_i as the center and r as the radius. The distance from minutiae $M_j(x_j, y_j, \theta_j)$ $(M_j \in MT_f, j \neq i)$ in MT_f to this circle is:

$$dis\_to\_c_{M_j,c(i,r)} = \left| \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} - r \right| \tag{4.6}$$

when $r = 0$, $dis\_to\_c_{M_j,c(i,r)}$ is the Euclidean distance between M_i and M_j.

Then, we find the two minutiae with the two lowest distances to the circle c. These three minutiae points form a triplet $T(m_1, m_2, m_3)$, where $m_1(x_1, y_1, \theta_1) = M_i$ as the first point, $m_2(x_2, y_2, \theta_2)$ and $m_3(x_2, y_2, \theta_2)$ are the minutiae with the lowest and second lowest distance to the circle c respectively. Figure 11 shows an example of deriving a triplet from a minutiae template.

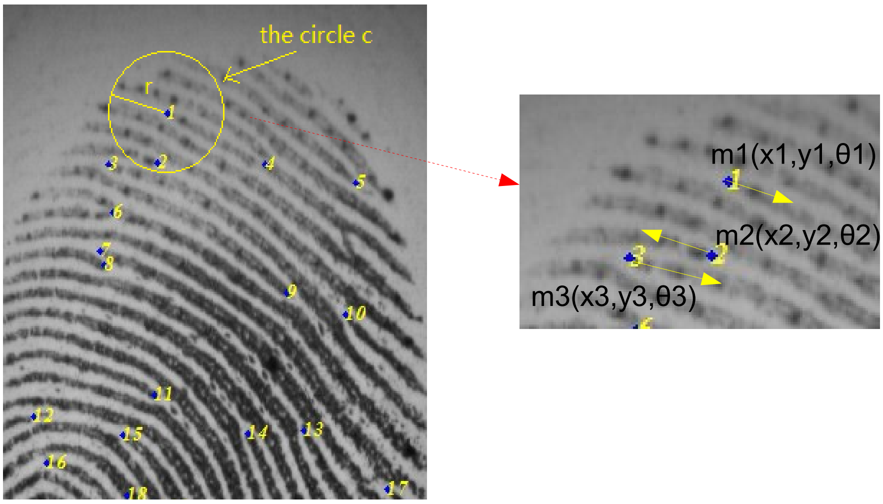


Figure 11: An example of deriving a triplet from a minutiae template. On the left side of the figure, we define the circle c to form a triplet for the minutiae 1. On the right side of the figure, it is the triplet we form for the minutiae 1.

We form a triplet for each of the minutiae in this template, so that N_{MT_f} triplets are found. After forming a triplet, we derive features from the triplet. Given a triplet $T(m_1, m_2, m_3)$, we define $\vec{V}_1$, $\vec{V}_2$, $\vec{V}_3$ as $\vec{v}_{12}$, $\vec{v}_{23}$, $\vec{v}_{31}$ according to the Equation 4.1 respectively. The features derived from a triplet are listed as follows:

1. l_i, which is the length of the vector $\vec{V_i}$. According to Equation 4.3, we have:

$$\begin{aligned} l_1 &= ed_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \\ l_2 &= ed_{23} = \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} \\ l_3 &= ed_{31} = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2} \end{aligned} \tag{4.7}$$

2. ave_l, which is the average value of l_1, l_2 and l_3

$$ave_l = \frac{\sum_{i=1}^{3} l_i}{3} \tag{4.8}$$

3. α_{ij}, which is the angle required to rotate the direction of $\vec{V_j}$ anticlockwise to the ridge orientation of minutiae m_i. According to Equation 4.4, we have:

$$\alpha_{ij} = angle(\theta_i, dir(\vec{V_j})) \tag{4.9}$$

4. β_{ij}, which is the angle required to rotate the ridge orientation of m_j anticlockwise to the ridge orientation of m_i. According to Equation 4.4, we have:

$$\beta_{ij} = angle(\theta_i, \theta_j) \tag{4.10}$$

Figure 12 illustrates all the features derived from a triplet. A triplet is illustrated in this figure, the three dots represent the three minutiae m_1, m_2 and m_3. In the figure, the arrows point from one minutiae to another represent the vector formed by the two minutiae. The arrows which have a minutiae as the initial point but do not have any minutiae as the terminal point represent the ridge orientation of the initial-point minutiae. 13 features are derived from a triplet: l_1, l_2, l_3, ave_l, β_{12}, β_{13}, β_{23}, α_{11}, α_{13}, α_{21}, α_{22}, α_{32}, α_{33}.

4.3 Requirements

The 13 triplet-based features are extracted from a triplet, then we do statistical analysis for these features to find a number of robust features that are reliable to be used as transformation parameters in our template protection algorithm. Both accuracy and privacy protection performance should be concerned. The requirements for the reliable features are as follows:

1. The reliable features should have relatively high entropy. Low feature entropy means that this feature is not distinctive enough. Low feature entropy results in low template entropy. Low template entropy makes it easy for the attacker to guess the template.

2. The reliable features should be independent. Suppose there are two features X and Y, the joint entropy of X and Y is:

$$H(X, Y) = H(X) + H(Y) - I(X, Y) \tag{4.11}$$

   where $I(X, Y)$ is the mutual information of X and Y.

   When X and Y are independent, $I(X, Y)$ is 0. If X and Y are not independent, H(X,Y) will be smaller than the case that X and Y are independent, since $I(X, Y)$ is above zero. It will reduce the template entropy and make it easier for the attacker to guess the template.

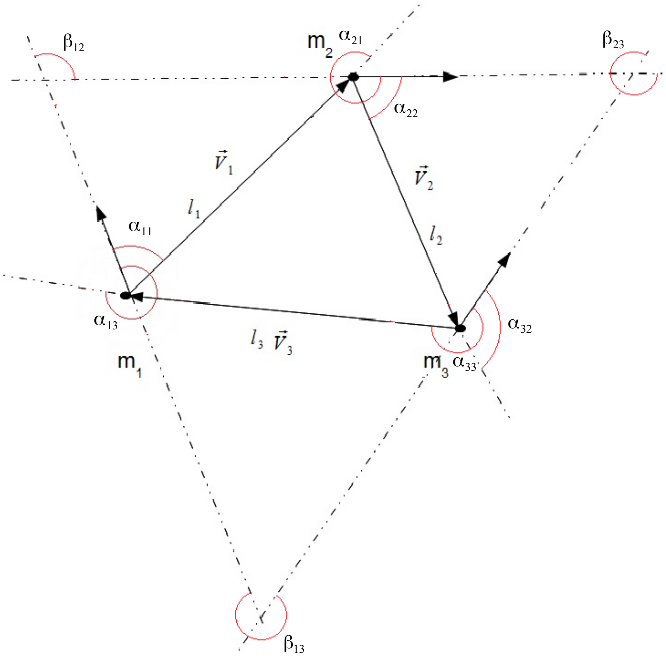


Figure 12: Triplet-based features extraction illustration.

3. The features are preferred if they are invariant to the order of the three minutiae in the triplet where they are derived from. The intra-class variance exits among the minutiae templates obtained from multiple acquisitions. Given a triplet $T(m_1, m_2, m_3)$, if we find the same triplet in another template of the same finger, the order of m_2 and m_3 may change because of the non-linear distortion due to the different finger angles and pressures put on the finger. If a feature is invariant to the order of the minutiae in the triplet, the intra-class variance of this feature can be reduced.

4.4 Feature analysis

FVC2002 DB2_A database[44] was used for the feature analysis work. The plain text minutiae templates were extracted by the fingerprint extractor from NeuroTechnology[45]. We set $r = 0$ and $r = 50$ respectively to derive the triplets. For each r value, all triplets were found in all the minutiae templates in the database and the 13 features presented in Section 4.2 were derived from all the triplets. The probability density distribution, entropy and dynamic range of these features and their correlations under the two cases of r are analyzed respectively. Our goal is to find the features that meet the requirements presented in Section 4.3 to use them as transformation parameters in our template protection algorithm.

4.4.1 Distribution

Figure 14 and 15 illustrate the histogram of the probability density distribution for the features when $r = 0$. Figure 16 and 17 illustrate the histogram of the probability density distribution for the features when $r = 50$. Table 1 lists the maximum values, minimum values and entropies for the features when $r = 0$ and $r = 50$ respectively. To calculate the entropy of a feature F, we divide the dynamic range of the feature space of F by the size of 5 and obtain m_0 segments, as illustrated in Figure 13. The entropy of F is:

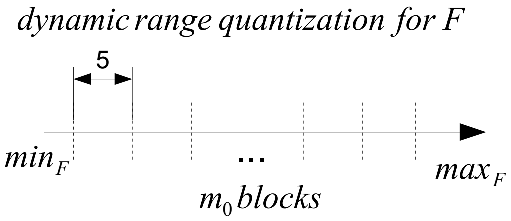


Figure 13: Dynamic range quantization for F. max_F and min_F are the maximum and minimum values for F respectively.

$$H(F) = -\sum_{k=1}^{m_0} p(k \cdot 5 - 5 \leq f < k \cdot 5) \cdot \log p(k \cdot 5 - 5 \leq f < k \cdot 5) \qquad (4.12)$$

Table 1: The maximum value, minimum value and entropy of each feature

	Minimum	Maximum	Entropy	Minimum	Maximum	Entropy
	$r = 0$			$r = 50$		
l_1	1.0000	278.9158	3.3414	4.000	278.9158	2.3222
l_2	3.1623	289.3372	3.6537	5.3852	289.3372	3.0016
l_3	1.0000	246.1097	4.1258	1.0000	246.1097	4.4415
ave_l	5.3333	220.3352	3.5643	9.6095	220.3352	3.1756
β_{12}	0	359	4.9734	0	359	5.5279
β_{13}	0	359	5.1868	0	359	5.5539
β_{23}	0	359	5.3835	0	359	5.7644
α_{11}	0	359.9946	6.1362	0	359.9946	6.1619
α_{13}	0	359.9946	6.1561	0	359.9946	6.1587
α_{21}	0	359.9910	6.1397	0	359.9946	6.1599
α_{22}	0	359.9965	6.1630	0	359.9975	6.1607
α_{32}	0	359.9910	6.1608	0	359.9987	6.1581
α_{33}	0	359.9965	6.1640	0	359.9965	6.1632

From the figures, we can see that for both $r = 0$ and $r = 50$, α_{11}, α_{13}, α_{21}, α_{22}, α_{32}, α_{33} follow approximate uniform distribution within the range of $[0, 360^\circ)$. β_{12} ,β_{13} and β_{23} follow non-typical distributions within the range of $[0, 360^\circ)$, and are not distributed equally within their ranges. l_1, l_2, l_3 and ave_l have narrower ranges and smaller entropies compared to the other angle-features. When $r = 0$, l_1, l_2, l_3 and ave_l follow normal distributions with a narrower range than the other angle-features. When $r = 50$, the distribution of l_1 and ave_l are more centralized and the entropy of l_1 is lower than the case of $r = 0$.

4.4.2 Correlation

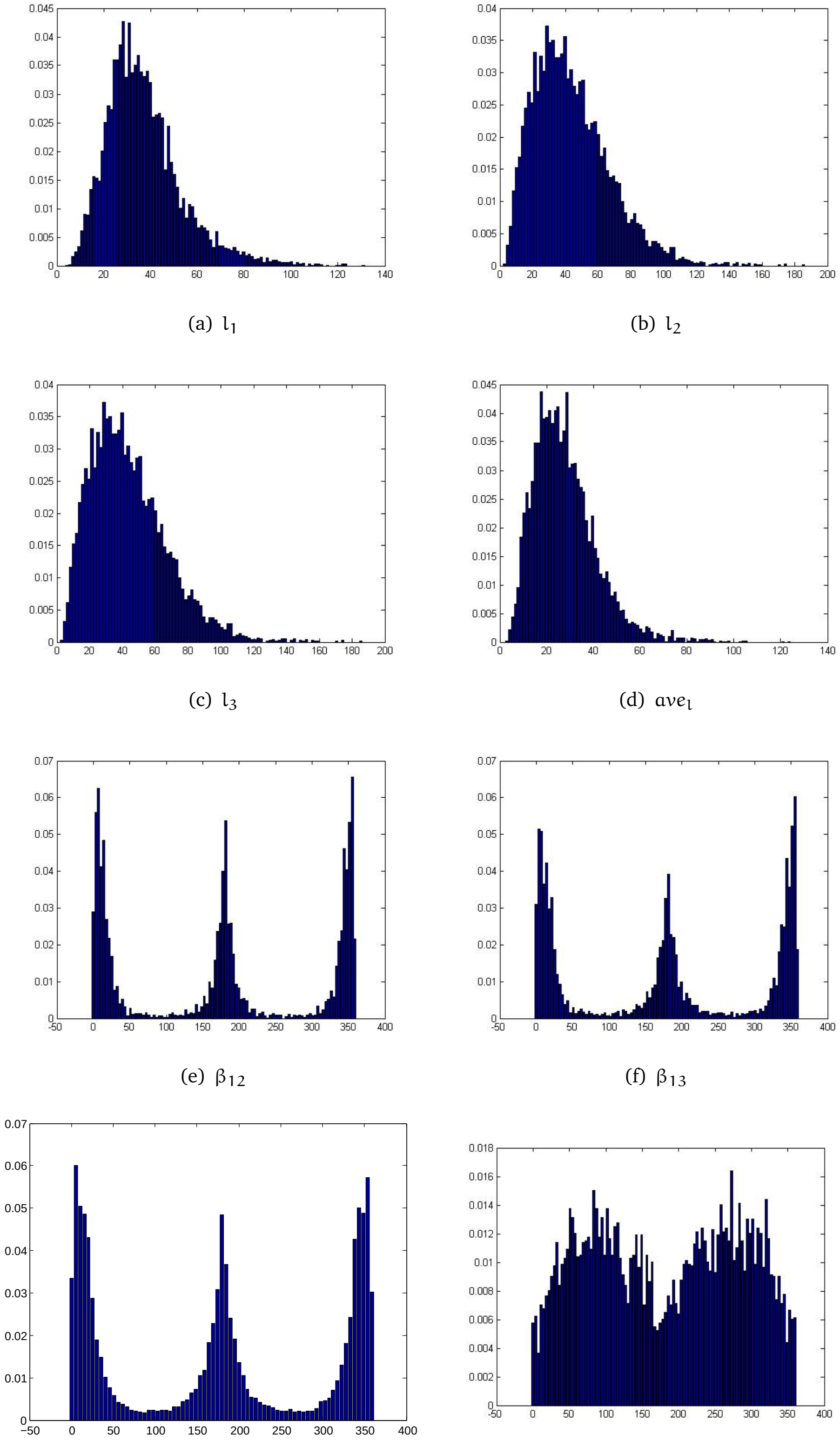
We analyze the correlation among the 13 features using Pearson's correlation coefficient [46] and mean value as the measure of expectation. Pearson correlation coefficient falls between [-1, 1]. The coefficient that is closed to -1, 1 and 0 indicates high negative correlation, high positive correlation and weak correlation respectively.

Suppose Feature X and Y are two features from from the 13 features. The Pearson's correlation coefficient of X and Y is:

$$\rho_{X,Y} = \frac{\frac{1}{N} \cdot \sum_{i=1}^{N} (x_i - \mu_X)(y_i - \mu_Y)}{\sigma_X \cdot \sigma_Y} \tag{4.13}$$

where μ_X and μ_Y are the mean value of X and Y respectively, σ_X and σ_Y are the standard deviations of X and Y respectively. N is the number of the samples for the X and Y.

Table 2 and 3 list the Pearson's linear correlation coefficients among these 13 features when $r = 0$ and $r = 50$ respectively. The correlation results combine physiological correlation and pressure-distortion (the distortion of minutiae samples due to the pressure put on the finger during the data capture process) correlation. From the table, we can see that when $r = 0$, l_1, l_2, l_3 and ave_l are highly correlated with one another. When $r = 50$, the correlations among the features are reduced compared to the case $r = 0$.



(a) l_1

(b) l_2

(c) l_3

(d) ave_l

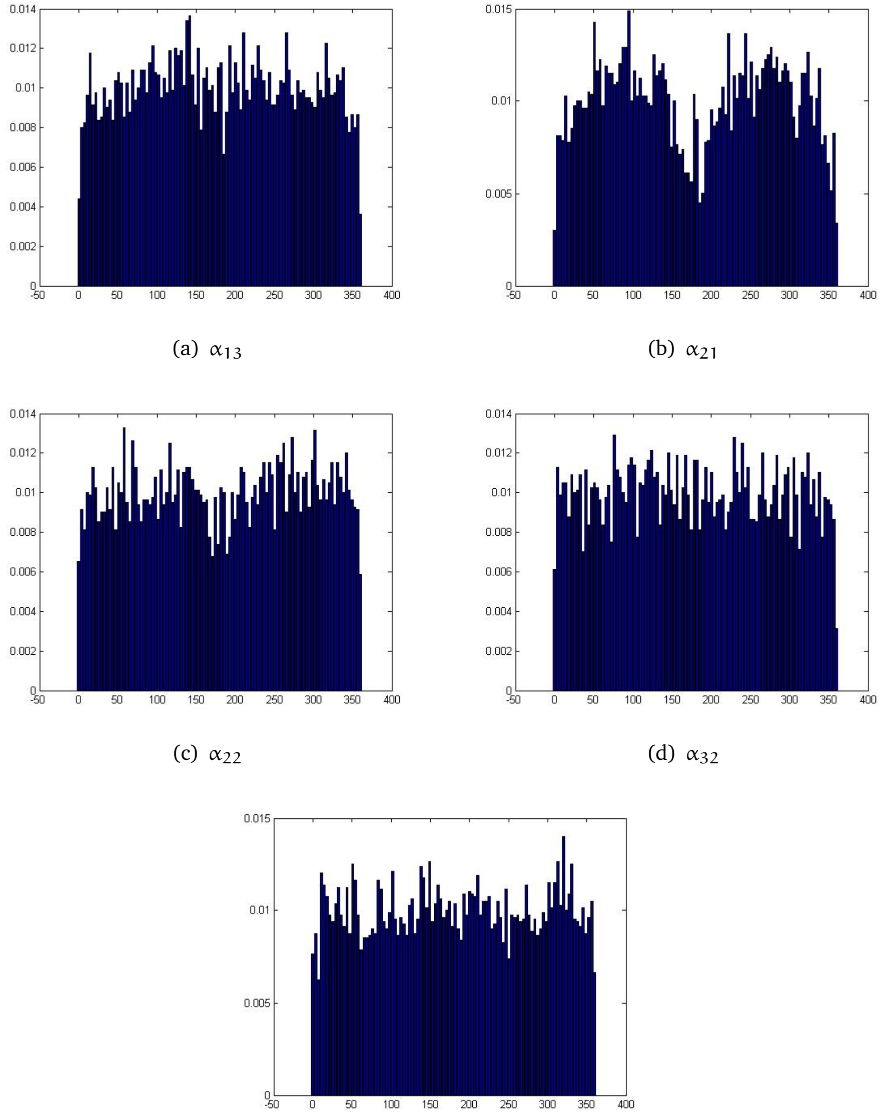
(e) β_{12}

(f) β_{13}

(g) β_{23}

(h) α_{11}

Figure 14: The probability density distribution of l_1,l_2,l_3,ave_l,β_{12},β_{13} ,β_{23} and α_{11} when $r = 0$



(a) α_{13}

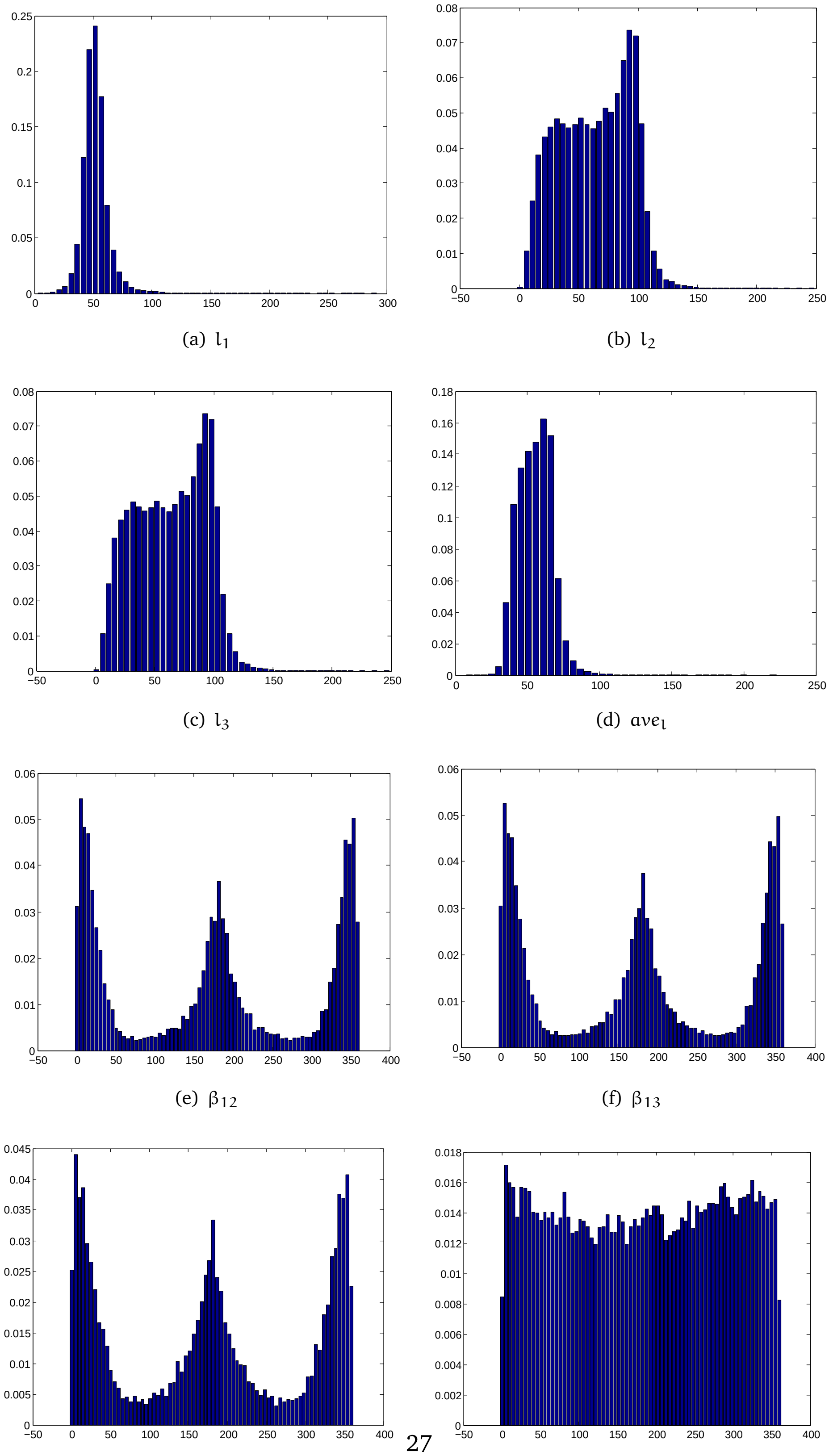
(b) α_{21}

(c) α_{22}

(d) α_{32}

(e) α_{33}

Figure 15: The probability density distribution of α_{13}, α_{21}, α_{22},α_{32}, α_{33} when $r = 0$



(a) l_1

(b) l_2

(c) l_3

(d) ave_l

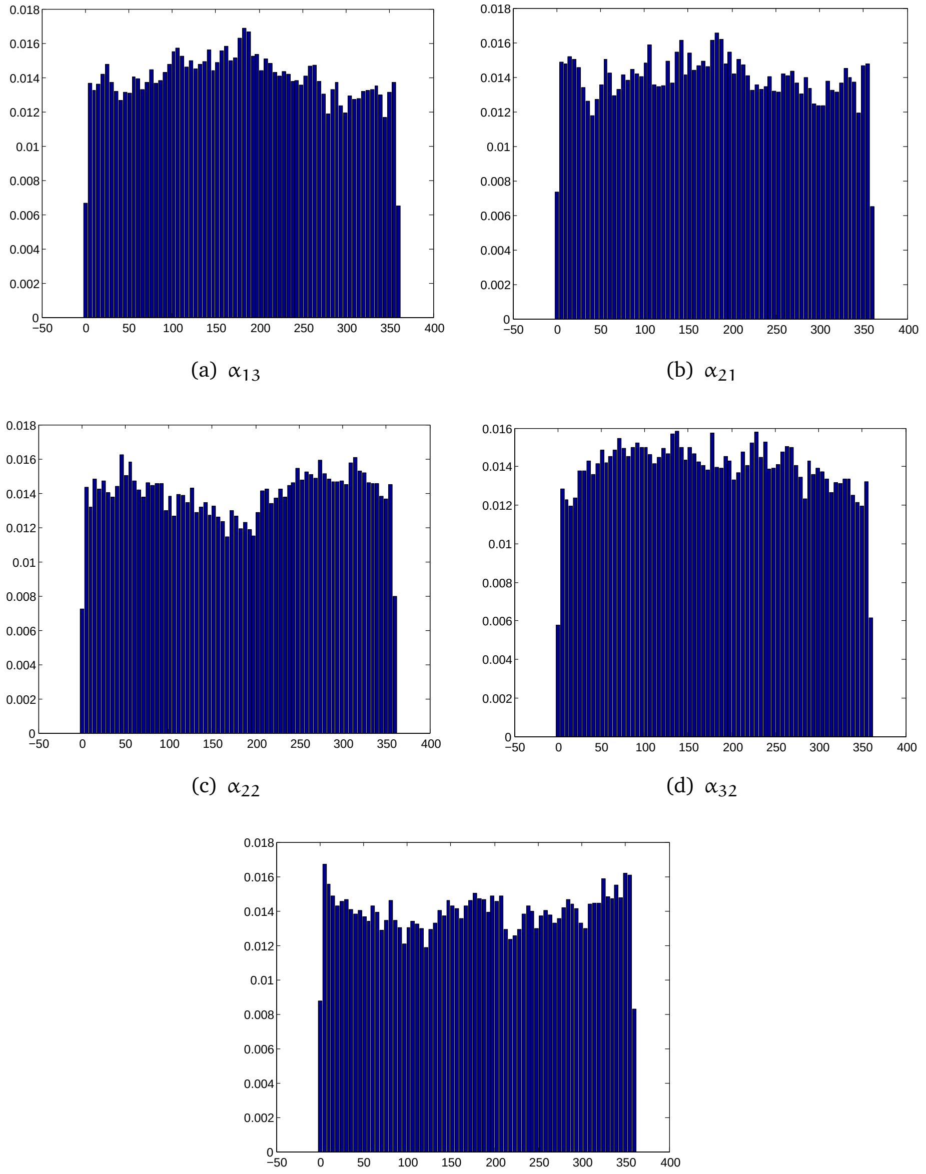
(e) β_{12}

(f) β_{13}

(g) β_{23}

(h) α_{11}

Figure 16: The probability density distribution of l_1,l_2,l_3,ave_l,β_{12},β_{13} ,β_{23} and α_{11} when $r = 50$



(a) α_{13}

(b) α_{21}

(c) α_{22}

(d) α_{32}

(e) α_{33}

Figure 17: The probability density distribution of α_{13}, α_{21}, α_{22},α_{32}, α_{33} when $r = 50$

Table 2: The correlations among the triplet-based features when $r = 0$.

	l_1	l_2	l_3	ave_l	β_{12}	β_{13}	β_{23}	α_{11}	α_{13}	α_{21}	α_{22}	α_{32}	α_{33}
l_1	1.0000	0.7462	0.5126	0.8274	-0.0097	-0.0042	0.0020	0.0070	-0.0081	0.0160	-0.0083	-0.0061	-0.0049
l_2	0.7462	1.0000	0.5931	0.8847	-0.0105	-0.0025	0.0016	-0.0001	-0.0065	0.0207	-0.0058	0.0021	-0.0034
l_3	0.5126	0.5931	1.0000	0.8646	-0.0065	0.0072	0.0114	-0.0007	0.0011	0.0051	-0.0053	-0.0011	0.0059
ave_l	0.8274	0.8847	0.8646	1.0000	-0.0100	0.0014	0.0068	0.0017	-0.0043	0.0150	-0.0073	-0.0016	0.0002
β_{12}	-0.0097	-0.0105	-0.0065	-0.0100	1.0000	0.0094	-0.2312	-0.4092	-0.0062	-0.4356	0.2046	0.1346	-0.0229
β_{13}	-0.0042	-0.0025	0.0072	0.0014	0.0094	1.0000	0.3569	-0.0062	0.3361	-0.0384	-0.1408	-0.2280	0.2889
β_{23}	0.0020	0.0016	0.0114	0.0068	-0.2312	0.3569	1.0000	0.1331	0.1890	0.1324	-0.3147	-0.3271	0.2348
α_{11}	0.0070	-0.0001	-0.0007	0.0017	-0.4092	-0.0062	0.1331	1.0000	-0.0337	0.3602	-0.1167	-0.0951	-0.0065
α_{13}	-0.0081	-0.0065	0.0011	-0.0043	-0.0062	0.3361	0.1890	-0.0337	1.0000	-0.0032	-0.0510	-0.1915	0.3761
α_{21}	0.0160	0.0207	0.0051	0.0150	-0.4356	-0.0384	0.1324	0.3602	-0.0032	1.0000	-0.2812	-0.1398	0.0512
α_{22}	-0.0083	-0.0058	-0.0053	-0.0073	0.2046	-0.1408	-0.3147	-0.1167	-0.0510	-0.2812	1.0000	0.3223	-0.0860
α_{32}	-0.0061	0.0021	-0.0011	-0.0016	0.1346	-0.2280	-0.3271	-0.0951	-0.1915	-0.1398	0.3223	1.0000	-0.4251
α_{33}	-0.0049	-0.0034	0.0059	0.0002	-0.0229	0.2889	0.2348	-0.0065	0.3761	0.0512	-0.0860	-0.4251	1.0000

Table 3: The correlations among the triplet-based features when $r = 50$.

	l_1	l_2	l_3	ave_l	β_{12}	β_{13}	β_{23}	α_{11}	α_{13}	α_{21}	α_{22}	α_{32}	α_{33}
l_1	1.0000	0.2157	0.0719	0.3432	0.0060	-0.0007	-0.0028	-0.0008	-0.0043	0.0037	-0.0035	0.0017	-0.0060
l_2	0.2157	1.0000	0.1646	0.5139	-0.0030	0.0005	-0.0003	-0.0060	0.0042	0.0109	-0.0059	0.0058	-0.0037
l_3	0.0719	0.1646	1.0000	0.9059	0.0056	0.0082	0.0112	-0.0049	0.0106	-0.0016	0.0054	-0.0048	0.0045
ave_l	0.3432	0.5139	0.9059	1.0000	0.0049	0.0069	0.0087	-0.0063	0.0093	0.0031	0.0019	-0.0017	0.0013
β_{12}	0.0060	-0.0030	0.0056	0.0049	1.0000	0.0463	-0.2197	-0.2356	0.0406	-0.2398	0.1709	0.0854	0.0172
β_{13}	-0.0007	0.0005	0.0082	0.0069	0.0463	1.0000	0.2308	-0.0421	0.2349	-0.0352	-0.0810	-0.1368	0.2168
β_{23}	-0.0028	-0.0003	0.0112	0.0087	-0.2197	0.2308	1.0000	0.0887	0.0804	0.1139	-0.2261	-0.2512	0.1149
α_{11}	-0.0008	-0.0060	-0.0049	-0.0063	-0.2356	-0.0421	0.0887	1.0000	-0.0871	0.2726	-0.0706	-0.0964	0.0004
α_{13}	-0.0043	0.0042	0.0106	0.0093	0.0406	0.2349	0.0804	-0.0871	1.0000	-0.0190	-0.0494	-0.1183	0.2647
α_{21}	0.0037	0.0109	-0.0016	0.0031	-0.2398	-0.0352	0.1139	0.2726	-0.0190	1.0000	-0.3467	-0.1004	-0.0215
α_{22}	-0.0035	-0.0059	0.0054	0.0019	0.1709	-0.0810	-0.2261	-0.0706	-0.0494	-0.3467	1.0000	0.2277	-0.0493
α_{32}	0.0017	0.0058	-0.0048	-0.0017	0.0854	-0.1368	-0.2512	-0.0964	-0.1183	-0.1004	0.2277	1.0000	-0.3522
α_{33}	-0.0060	-0.0037	0.0045	0.0013	0.0172	0.2168	0.1149	0.0004	0.2647	-0.0215	-0.0493	-0.3522	1.0000

5 The Proposed interoperable minutiae template protection method–Renewable and Interoperable Minutiae Encoder

In this chapter, we introduce RIME (Renewable and Interoperable Minutiae Encoder) algorithm. It is a feature transformation method that uses triplet-based features as the transformation parameters and random projection as the transformation function.

5.1 Functions

This section gives the functions used in RIME. Suppose that there is a plain text minutiae template MT_f from finger f. The functions are as the follows:

1. Triplet generation function:

$$[T1, T2, T3, ndis_1, ndis_2, ndis_3] = TripletGen(MT_f, i, r) \tag{5.1}$$

First, this function defines the i^{th} minutiae $M_i(x_i, y_i, \theta_i)$ in MT_f as the first minutiae m_1 in the triplets T1, T2 and T3. Then, this function defines a circle c with $M_i(x_i, y_i, \theta_i)$ as the center and r as the radius. Among all $M_j \in MT_f, j \neq i$, this function finds the minutiae with the first, second and third nearest distance to c.

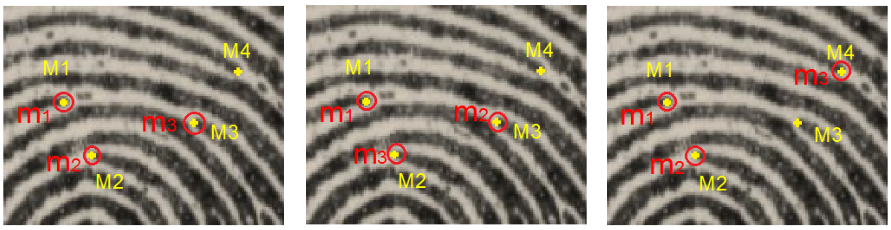
$dis\_to\_c_{M_j,c(i,r)}$, which is the distance from minutiae $M_j(x_j, y_j, \theta_j)$ $(M_j \in MT_f, j \neq i)$ to c, is defined as:

$$dis\_to\_c_{M_j,c(i,r)} = \mid \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} - r \mid \tag{5.2}$$

This function takes the square of the k^{th} $(k \in \{1, 2, 3\})$ nearest distance as $ndis_k$. The minutiae with the nearest distance and the second nearest distance to c are presented as m_2 and m_3 respectively in T1. The minutiae with the second nearest distance and the nearest distance to c are presented as m_2 and m_3 respectively in T2, and the minutiae with the nearest distance and the third nearest distance to c are presented as m_2 and m_3 respectively in T3.

Figure 18 illustrates an example of forming three triplets for a given minutiae. In the figure, given minutiae M1, suppose $r = 0$, we first find minutiae M2, M3 and M4 which are the first, second and third nearest minutiae to M1 respectively, then we from T1 as $T1(M1, M2, M3)$, T2 as $T2(M1, M3, M2)$ and T3 as $T3(M1, M2, M4)$.

Given a triplet $T(m_1, m_2, m_3)$ from a minutiae template, if we find the same triplet in another template of the same finger, the order of m_2 and m_3 may change because of the non-linear distortion due to the different finger angles and pressures put on the finger. m_3 may be the minutiae with the third nearest distance to m_1 as well. Thus, we take $T2$ and $T3$ as backups for $T1$.



T1 T2 T3

Figure 18: An example of forming three triplets for the given minutiae $M1$

2. Parameter generation function:

$$par = GetPar(T, mean, std) \tag{5.3}$$

This function calculates the parameter vector $par = [par_1, par_2, ..., par_k]^T$ used as transformation parameters in RIME. Suppose that k features chosen from the 13 features introduced in the Section 4.2 are derived from the triplets. Different combinations of features can be chosen. The inputs of the function $GetPar$ are a triplet T, a vector $mean$ which is a vector of the mean values of the chosen k features and a vector std which is a vector of the standard deviation values of the chosen k features. For the triplet T, we derive a feature vector v according to the chosen k features. $feature_i$ in the vector v is the value of the i^{th} feature, and the i^{th} rows in the $mean$ and the std are the mean value and standard deviation value for the i^{th} feature respectively. The calculation of the features is presented in Section 4.2.

$$v = \begin{bmatrix} feature_1 \\ feature_2 \\ ... \\ feature_k \end{bmatrix} \tag{5.4}$$

$$mean = \begin{bmatrix} mean_1 \\ mean_2 \\ ... \\ mean_k \end{bmatrix} \tag{5.5}$$

$$std = \begin{bmatrix} std_1 \\ std_2 \\ ... \\ std_k \end{bmatrix} \tag{5.6}$$

par is the output of the $GetPar$ function. It is the transformation parameters used in RIME. It is calculated as:

$$par = (v - mean)./std \tag{5.7}$$

In this step, if $mean$ and std are set as the mean values and standard deviation values of the features, the features will be normalized. But $mean$ and std are not mandatorily to be set as the mean values and the standard deviation values. We can set $mean$ as $[0 \quad 0 \quad ... \quad 0]^T$ and std as $[1 \quad 1 \quad ... \quad 1]^T$, then the original values of the features will be kept. They can be set as other values to give different weights to the features as well.

3. Transformation function:

$$PT = transformation(par, Key_f, Coefficient) \tag{5.8}$$

The inputs of this function are a parameter vector par generated by the function $GetPar$, a constant $Coefficient$ and Key_f which is the transformation key for the finger f. The output of this function is a three dimensional vector $PT = [PTx, PTy, PT\theta]^T$. We regard PTx, PTy and $PT\theta$ as the x-coordinate, y-coordinate and ridge orientation of minutiae, thus, this vector can be seen as minutiae. The protected template of the plain text template MT_f is composed by such vectors thus it is compliant to ISO minutiae format. Key_f is a random generated $3 \times k$ orthonormal matrix whose columns and rows are orthogonal unit vectors. It is the user-specific transformation key for the finger f. $Coefficient$ is a constant for adjusting the range of PT. PT is calculated as:

$$\begin{bmatrix} PTx \\ PTy \\ PT\theta_{pre} \end{bmatrix} = Key_f \cdot par \cdot Coefficient \tag{5.9}$$

$$PT\theta = PT\theta_{pre} \pmod{360} \tag{5.10}$$

$$PT = \begin{bmatrix} PTx \\ PTy \\ PT\theta \end{bmatrix} \tag{5.11}$$

5.2 Detailed procedure of RIME

This section gives the procedure of RIME in details. Given a finger f, MT_f is a ISO standard minutiae template from the finger f. There is N_{MT_f} minutiae in MT_f. k features are chosen from l_1, l_2, l_3, ave_l, β_{12}, β_{13}, β_{23}, α_{11},α_{13}, α_{21}, α_{22}, α_{32} and α_{33} as the features that are derived from the triplets. A $3 \times k$ orthonormal matrix Key_f is generated as the transformation key for the finger f. Then we set the parameters r, $mean$, std, $Coefficient$ and $DistTH$. The RIME algorithm is:

```
$n \leftarrow 0$
for $i = 1 \rightarrow N_{MT_f}$ do
    $n \leftarrow n + 1$
    $[T1, T2, T3, ndis_1, ndis_2, ndis_3] \leftarrow TripletGen(MT_f, i, r)$
    $par_n \leftarrow GetPar(T1, mean, std)$
    $PT_n \leftarrow transformation(par_n, Key_f, Coefficient)$
    if $ndis_2 - ndis_1 \leq DistTH$ then
        $n \leftarrow n + 1$
```

```
            $par_n \leftarrow GetPar(T2, mean, std)$
            $PT_n \leftarrow transformation(par_n, Key_f, Coefficient)$
        end if
        if $ndis_3 - ndis_2 \leq DistTH$ then
            $n \leftarrow n + 1$
            $par_n \leftarrow GetPar(T3, mean, std)$
            $PT_n \leftarrow transformation(par_n, Key_f, Coefficient)$
        end if
    end for
```

In this algorithm, PT is the protection template generated from MT_f. par_n and PT_n are the n^{th} transformation parameter vector and the n^{th} point in PT respectively. DistTH is used as a threshold. In the triplet generation function $[T1, T2, T3, ndis_1, ndis_2, ndis_3] = TripletGen(MT_f, i, r)$, $ndis_k$ is the square of the distance between the k^{th} $(k \in \{1, 2, 3\})$ nearest minutiae to the i^{th} minutiae in MT_f. If $ndis_2 - ndis_1 \leq DistTH$, with regard to the intra-class variability of multiple fingerprint samples from one finger, we extract features and generate a point in PT from T2. If $ndis_3 - ndis_2 \leq DistTH$, we generate a point in PT from T3.

PT is formatted as a set of points with three dimensions: PTx, PTy and PTθ. For each point, we regard it as a minutiae of which the x-coordinate is PTx, the y-coordinate is PTy and the angle is PTθ. Thus, PT is compliant to ISO minutiae template format.

6 Experiments

In this chapter, we evaluate the accuracy performance of RIME with choosing different features and different parameter settings. The experiments were done on the FVC2002 DB2_A database[44]. The plaint text minutiae templates were extracted by the minutiae extractor from NeuroTechnology[45]. Bozorth3 from NIST[47] and Verifinger comparator from NeuroTechnology[45] were used as the comparators. EER was used as one of the accuracy indicators. When using Verifinger, we got 0 FMR, thus the FNMR when FMR is zero was used as a accuracy indicator as well.

There are 100 fingers and 8 samples for each finger in the database. We used the first sample as the reference and the other 7 samples to do the verification for each finger. When using plain text minutiae templates for fingerprint verification and Bozorth3 as the comparator, the EER is 1.97. Figure19 illustrated the DET curve.

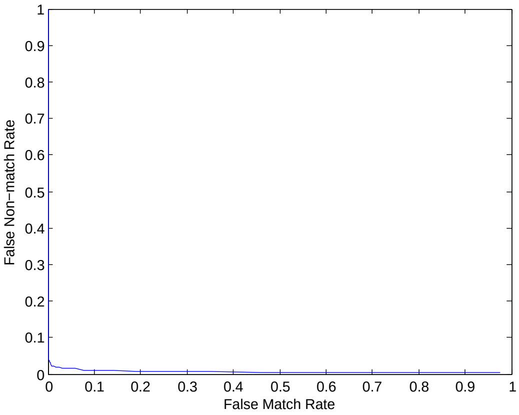


Figure 19: DET curve of using plain text minutiae templates for fingerprint verification.

We test the accuracy performance of RIME with four different sets of features, meaning four different feature vectors defined in the Equation 5.4, to generate the transformation parameters. These feature vectors are:

1.

$$v_1 = \begin{bmatrix} l_1 \\ \alpha_{11} \\ \alpha_{21} \end{bmatrix} \tag{6.1}$$

When using this feature vector, only the first two minutiae in the triplet are used.

2.

$$v_2 = \begin{bmatrix} \alpha_{11} \\ \alpha_{22} \\ \alpha_{33} \end{bmatrix} \tag{6.2}$$

The three features in this vector have relatively high entropies and low correlation among them.

3.

$$v_3 = \begin{bmatrix} \alpha_{11} \\ \alpha_{22} \\ \alpha_{33} \\ \beta_{23} \end{bmatrix} \tag{6.3}$$

4.

$$v_4 = \begin{bmatrix} \alpha_{11} \\ \alpha_{22} \\ \alpha_{33} \\ \text{ł}_1 \\ \text{ł}_3 \end{bmatrix} \tag{6.4}$$

v_3 and v_4 are obtained by adding features which have lower entropies to v_2.

Three sets of parameter settings were used for testing:

1. $r = 0$, $mean = [0 \quad 0 \quad 0]^T$ and $std = [1 \quad 1 \quad 1]^T$. In this case, the original feature values are used as transformation parameters.
2. $r = 0$, $mean$ and std are set as the mean values and standard deviation values for the features. In this case, the feature values are normalized and then used as transformation parameters.
3. $r = 50$, $mean = [0 \quad 0 \quad 0]^T$ and $std = [1 \quad 1 \quad 1]^T$. In this case, the correlation among the features are reduced compared to the case $r = 0$.

$DistTH$ is set as 5 for all the experiments. The setting of $Coefficient$ depends on the feature values used as transformation parameters. The $Coefficient$ should be set bigger when using the normalized feature values as transformation parameters than using the original feature values to enlarge the range of the output, since the normalized feature values are much smaller than the original values.

Table 4, 5 and 6 summary the experiment results with 8 groups of parameter settings . The DET curve for the experiments with different settings are illustrated in Figure 20 and 21 with the corresponding setting serial number in the tables. When using v_2 to test the accuracy, Verfinger was used and obtained zero FMR, thus the FNMR when FMR is zero when using Veringer is only included in Table 5. From the experiments, we can see that $v_2 = [\alpha_{11}, \alpha_{22}, \alpha_{33}]^T$ obtained the best accuracy. When using $v3$ and $v4$, β_{23} or $l_1 andl_3$ is added into v_2, the accuracy is reduced, as shown in Table 6. β_{23}, $l_1 andl_3$ have lower entropy than the features in $v2$.

Table 4: Experiment results on database DB2_A

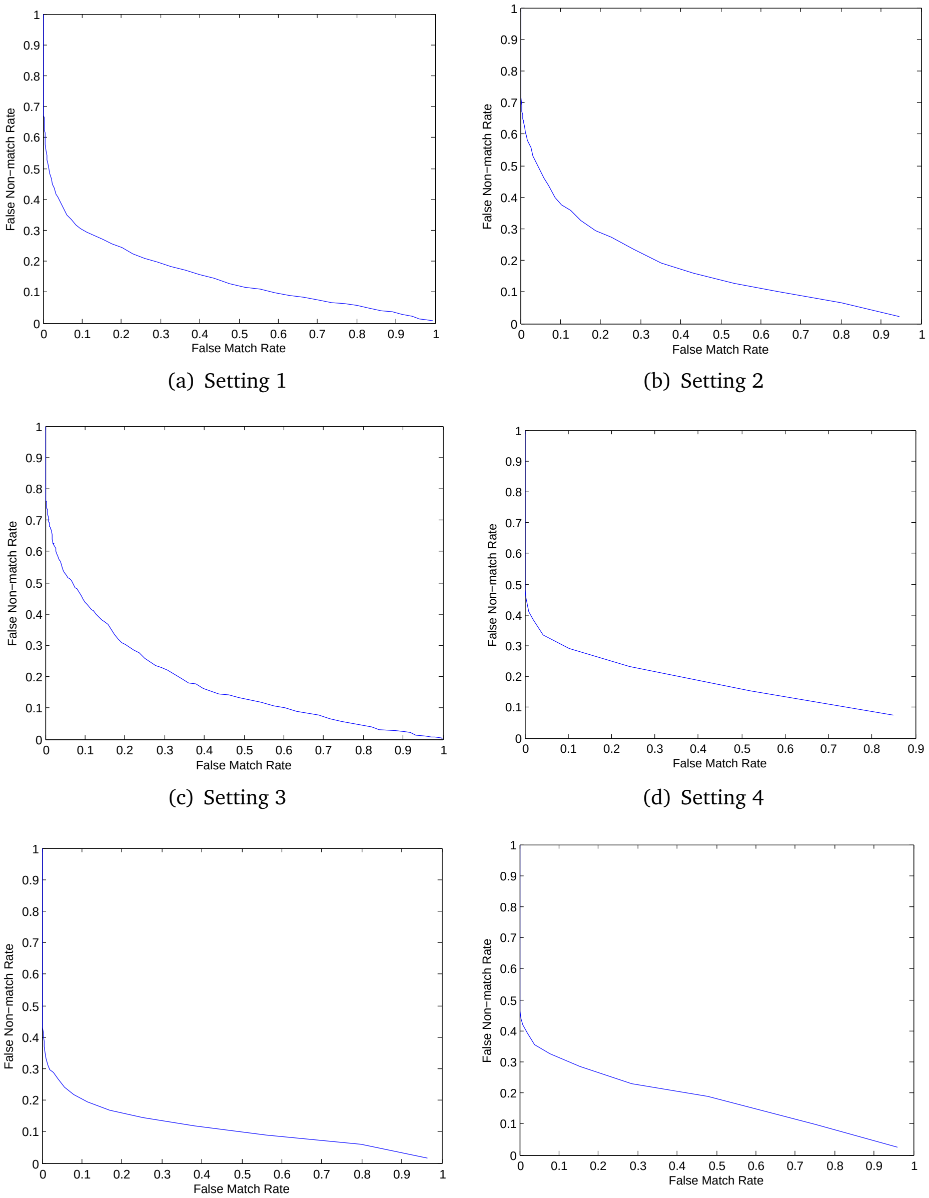
No.		Setting 1	Setting 2	Setting 3
Feature vector		$v_1 = [l_1, \alpha_{11}, \alpha_{21}]^T$		
Parameter settings	r	0	0	50
	DistTH	5	5	5
	Coefficient	1	100	1
	mean	$[0\ 0\ 0]^T$	$[31.58\ 181.2\ 180.3]^T$	$[0\ 0\ 0]^T$
	std	$[1\ 1\ 1]^T$	$[15.28\ 103.9\ 103.5]^T$	$[1\ 1\ 1]^T$
EER(%)	Bozorth3	22.38	25.43	25.56
FNMR when FMR=0(%)	Bozorth3	89.86	88.09	92.54

Table 5: Experiment results on database DB2_A

No.		Setting 4	Setting 5	Setting 6
Feature vector		$v2 = [\alpha_{11}, \alpha_{22}, \alpha_{33}]^T$		
Parameter settings	r	0	0	50
	DistTH	5	5	5
	Coefficient	1	80	1
	mean	$[0\ 0\ 0]^T$	$[180.3\ 181.3\ 180.1]^T$	$[0\ 0\ 0]^T$
	std	$[1\ 1\ 1]^T$	$[102.2\ 104.3\ 104.1]^T$	$[1\ 1\ 1]^T$
EER(%)	Bozorth3	21.57	16.11	23.77
FNMR when FMR=0(%)	Bozorth3	50.65	60.11	52.80
	Verifinger	33.63	39.42	35.17

Table 6: Experiment results on database DB2_A

No.		Setting 7	Setting 8
Feature vector		$v3 = [\alpha_{11}, \alpha_{22}, \alpha_{33}, \beta_{23}]^T$	$v4 = [\alpha_{11}, \alpha_{22}, \alpha_{33}, l_1, l_3]^T$
Parameter settings	r	0	0
	DistTH	5	5
	Coefficient	80	80
	mean	$[180.3\ 181.3\ 180.1\ 92.17]^T$	$[180.3\ 181.3\ 180.1\ 27.89\ 43.04]^T$
	std	$[102.2\ 104.3\ 104.1\ 49.58]^T$	$[102.2\ 104.3\ 104.1\ 14.57\ 23.92]^T$
EER(%)	Bozorth3	25.88	45.35
FNMR when FMR=0(%)	Bozorth3	67.86	99.28



(a) Setting 1

(b) Setting 2

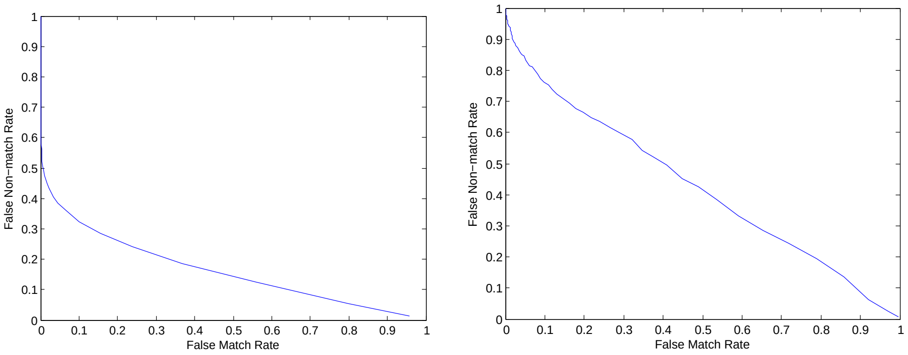
(c) Setting 3

(d) Setting 4

(e) Setting 5

(f) Setting 6

Figure 20: DET curves for the experiments with different settings. (1)



(a) Setting 1

(b) Setting 2

Figure 21: DET curves for the experiments with different settings (2)

7 Privacy Protection Performance Assessment

In this chapter, we analyse the privacy protection performance of RIME against brute force attack. Irreversibility and unlinkability are analyzed. The precondition of the analysis is that the attacker do not have access to the transformation key, and all parameters used in the algorithm are not known by the attacker. In Chapter 6, the feature vector $[\alpha_{11}, \alpha_{22}, \alpha_{33}]^T$ obtained the best performance, thus, in this chapter we give the methods for analyzing irreversibility and unlinkability of RIME but only present the irreversibility and unlinkability analysis results for using the feature vector $[\alpha_{11}, \alpha_{22}, \alpha_{33}]^T$ in RIME.

7.1 Irreversibility

Suppose that the attacker can use the biometric system to perform brute force attacks. The attacker uses a set of guessed values of the feature vector $v = [\mathtt{feature}_1, \mathtt{feature}_2, .., \mathtt{feature}_k]^T$ to generate the protected template and check if the generated template can match the target reference in the database, but no parameters used in RIME is known. The irreversibility can be evaluated by the number of possible guesses that the attacker need to make to guess the reference.

Suppose the features in v are independent. For $\mathtt{feature}_i$, as illustrated in Figure 22, we divide the dynamic range of the feature space by the size of 5 and obtain m_{0i} segments.

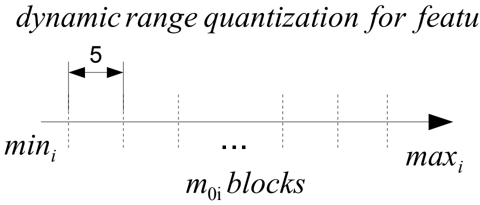


Figure 22: Dynamic range quantization for $\mathtt{feature}_i$. max_i and min_i are the maximum and minimum values for $\mathtt{feature}_i$ respectively.

The number of possible values for $\mathtt{feature}_i$ is:

$$n_i = m_{0i} \cdot \frac{H_{feature_i}}{H_{0i}} = m_{0i} \cdot \frac{H_{feature_i}}{\log(m_{0i})} \tag{7.1}$$

where $H_{feature_i}$ is the entropy of $\mathtt{feature}_i$. The calculation of the feature's entropy is presented in Equation 4.12.

Assume that if a probe matches the reference, there need to be at least 10 minutiae points match. The irreversibility for feature vector ν is:

$$\mathit{Irreversibility}_\nu = \log(10 \cdot \prod_{i=1}^{k} n_i) \quad (\mathrm{bits}) \tag{7.2}$$

Table 7 shows the irreversibility of RIME when using the feature vector $\nu = [\alpha_{11}, \alpha_{22}, \alpha_{33}]^T$ to generate transformation parameters.

Table 7: The irreversibility of RIME when using $\nu = [\alpha_{11}, \alpha_{22}, \alpha_{33}]^T$ to generate transformation parameters

Feature vector	$\nu = [\alpha_{11}, \alpha_{22}, \alpha_{33}]^T$
Irreversibility (/bits)	18

If the features in ν are correlated (for example, in the feature vector $[l_1, \alpha_{11}, \alpha_{21}]^T$, α_{11} and α_{21} are correlated to some degree according to Table 2 and 3), the irreversibility will be reduced since the joint entropy of the features will be smaller than the case that the features are independent, as analyzed in Section 4.3. Even though the attacker can find the feature values that are derived from minutiae triplets to generate the stored reference by brute force attack and get a feature-value template, there are still efforts needed to guess the original minutiae from the feature values. Thus, the irreversibility of reversing the original minutiae template from its protected template is higher than the irreversibility analyzed in this section.

7.2 Unlinkability

To analyze the unlinkability of RIME, we investigate the similarity and distance between two protected templates.

7.2.1 Similarity

Given a number of templates that are generated by RIME with the same chosen features and parameter settings, there are templates generated from the same finger and also different fingers. We use the first two dimensions which are the x-coordinate value and y-coordinate value of the templates to calculate the similarity. First, we obtain P which is the range of the first two dimensions of all the templates, and set a two dimensional space S of which the range covers P. P and S depend on the features and parameters used in RIME. Then, we divide the space S into blocks sized 5×5. For a protected template PT, we take the first two dimensions and count the number of the point that falls into each block of S, and generate a matrix M_{PT} for PT. The value of $M_{PT}\{m, n\}$ is the number of pints in the block at m^{th} row and n^{th} column of S. An example is illustrated in Figure 23.

The similarity of the template A and the template B is defined as:

$$\mathit{similarity}(A, B) = \sum_i \sum_j M_A\{i, j\} \cdot M_B\{i, j\} \tag{7.3}$$

The intra-similarity is defined as the similarity of the templates that are generated by the same finger but different keys. The inter-similarity is defined as he similarity of the templates that are

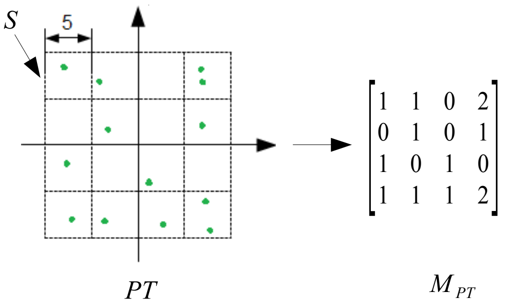


Figure 23: An example of generating the matrix for a protected template PT.

generated by the different fingers and different keys. We classify the intra-similarity values and inter-similarity values. Table 8 shows the classification results of using the Setting 4, 5 and 6 from Table 5. From Table 8, we can see that the EERs are around 50%, meaning that intra-similarity and inter-similarity are not distinguishable. It is hard to finger out if two protected templates are generated from the same finger or not by investigating the similarity between them.

Table 8: Classification of intra-similarity and inter-similarity

	Setting 4	Setting 5	Setting 6
Feature vector	$[\alpha_{11}, \alpha_{22}, \alpha_{33}]$		
EER(%)	49.85	51.01	53.15

7.2.2 Distance

We can also analyze the unlinkability of RIME from the aspect of the distance. Given two protected templates A and B, there are a and b points in the templates respectively. We denote PA_i and PB_i as the i^{th} three-dimensional points in A and B respectively. The distance from A to B is

$$dis(A,B) = \frac{1}{a} \cdot \sum_{i=1}^{a} min\{d(PA_i, PB_1), d(PA_i, PB_2), ..., d(PA_i, PB_b)\} \tag{7.4}$$

where $d(PA_i, PB_j)$ is the Euclidean distance between the points PA_i and PB_i.

The distance between A and B is:

$$Distance(A,B) = \frac{1}{2} \cdot (dis(A,B) + dis(B,A)) \tag{7.5}$$

Intra-distance is defined as the distance between two templates that generated from the same finger but different keys. Inter-distance is defined as the distance between two templates generated from two different fingers. We classify the intra-distance values and inter-distance values. The classification results of using the Setting 4, 5 and 6 from Table 5 are shown in Table 9. From Table 9, we can see that the intra-distance and the inter-distance are not distinguishable since the

Table 9: Classification of intra-distance and inter-distance

	Setting 4	Setting 5	Setting 6
Feature vector	$[\alpha_{11}, \alpha_{22}, \alpha_{33}]$		
EER (%)	49.52	53.04	49.32

EERs are around 50%, meaning it is hard to finger out if two protected templates are generated from the same finger or not by investigating the distance between them.

8 Two-Factor Authentication Using RIME

In this chapter, we discuss the application of RIME under two factor authentication scenario. Under this scenario, the accuracy of RIME is increased.

Two-factor authentication[48] is an authentication approach that requires the user to provide two or more of the three authentication factors:

1. something the user knows, like passwords or PIN.
2. something the user has, like a secure ID token or a cellphone.
3. What the user is, like biometric traits.

The authentication approach that combines three of the factors can be called three-factor authentication. Combing two passwords or two PIN could be considered as two-factor authentication, but only one factor is included.

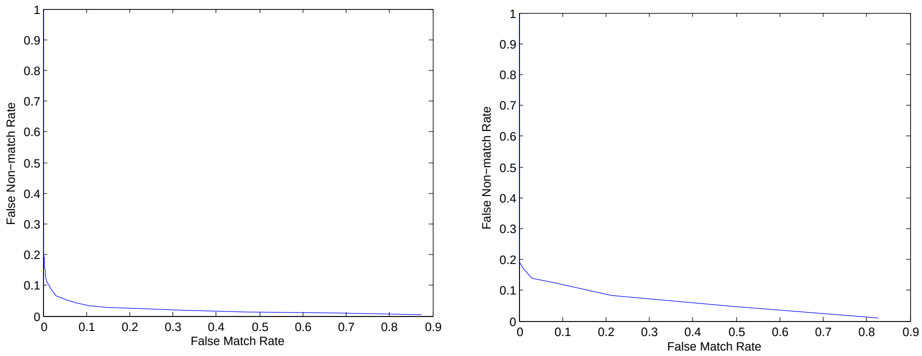
Under the template protection scenario, the user only provide his or her fingerprint, which is "what the user is". The user-specific transformation key is stored in the system. During verification, the imposter's minutiae template is transformed using the key of the identity that the imposter claims to be.

Different from the template protection scenario presented in the previous chapters, under two-factor authentication scenario, the user-specific transformation key is something that the user has. It can be stored in a secure ID token or a e-passport. During the verification, the user provides his or her fingerprint from finger f and his or her transformation key Key_f. Under this scenario, the imposter has to provide his or her fingerprint and transformation key, thus the imposter's minutiae template is transformed using his or her own key.

Table 10 shows the accuracy results of RIME under two factor authentication scenario using the Setting 1 from Table 4 and Setting 5 from Table 5. Figure8 illustrates the DET curves for the experiments with the corresponding setting serial numbers. Compare Table 10 with Table 4 and Table 5, we can see that the accuracy of RIME under two factor authentication scenario is increased compared to the case of template protection scenario using the same parameter setting.

Table 10: Two factor authentication using RIME

No.		Setting 1	Setting 5
Feature vector		$v1 = [l_1, \alpha_{11}, \alpha_{21}]^T$	$v2 = [\alpha_{11}, \alpha_{22}, \alpha_{33}]^T$
Parameter settings	r	0	0
	DistTH	5	5
	Coefficient	1	80
	mean	[0 0 0]	[180.3 181.3 180.1]
	std	[1 1 1]	[102.2 104.3 104.1]
EER(%)	Bozorth3	5.18	12.54



(a) Setting 1

(b) Setting 5

Figure 24: DET curves for the experiments under two factor authentication scenario

9 Discussion

9.1 Accuracy v.s. Interoperability

Observed from the experiment results in Chapter 6, on the FVC2002 DB2_A database, the accuracy performance is reduced after RIME is performed. The accuracy degradation exists. Which features we choose to generate transformation parameters in RIME has strong influence on the accuracy performance. Feature value normalization does not have significant influence on the accuracy.

The protected templates generated by RIME are compliant to ISO minutiae format. From our experiments, both of the minutiae template comparators, bozorth3 and Verifinger, can take the protected templates generated by RIME as inputs and output comparison scores. Thus, the interoperability among fingerprint verification systems that use ISO minutiae format templates is achieved. But there is trade-off between interoperability and accuracy.

9.2 FMR v.s. FNMR

The feature's entropy has an significant influence on the accuracy of RIME. The features that have lower entropy are less distinctive, such as l_1, l_2 and l_3, ave_l, β_{12}, β_{13} and β_{23}. These features cause higher FMR. The features that have higher entropy are more distinctive, such as α_{11}, α_{13}, α_{21}, α_{22}, α_{32}, α_{33}. These features increase FNMR. Overall, features with low entropy reduce the accuracy.

9.3 Security, diversity and revocability

In RIME, we use random projection to project the triplet based features onto a randomly selected orthogonal space. The random projection is invertible, thus, if an attacker gains access to the transformation key and the protected template, he can recover the original minutiae template. Hence, the security of the RIME is based on the secrecy of the key.

In the proposed method, the transformation key is user-specific. Hence, multiple templates for the same fingerprint can be generated by using different keys. This achieves diversity. If a protected template is compromised, it is easy to revoke the compromised template and generate a new template by using a different user-specific key. This enables revocability.

9.4 Why the performance is degraded to some degree

Except for the feature's entropy, the reliability of the points in the protected domain is also an important factor for the accuracy performance.

During verification, the minutiae comparator seeks for pairs of minutiae in the query and the reference that match. When using plain text templates for verification, if a pair of minutiae matches, it can be verified. In the protected domain, two points are generated from not less than two minutiae in the original template domain. For example, when using $[l_1, \alpha_{11}, \alpha_{21}]^T$ as the feature vector v, two to four original minutiae is needed to generate two points in the protected template. When using $[\alpha_{11}, \alpha_{22}, \alpha_{33}]^T$ as the feature vector v, three to six original minutiae is needed to generate two points in the protected template. Hence, the stability of two points in a protected template depends on the stability of not less than two raw minutiae. When using the protected template to do verification, the reliability of two points in the protected templates depends on the reliability of two to six minutiae in the original minutiae template. Hence, to match two points in a reference, for the query, if one of the needed minutiae is missed in triplets that are needed to generate the two points, the match will fail. This is an important reason for performance degradation.

Figure 25 shows an example of failure to match. On the left side there is the reference and on the right side there is the query. They are from the same finger and have six common minutiae. When using original minutiae template for verification, the six common minutiae can be verified. When using protected templates to do verification, PTR_1 and PTR_2 are two points in the protected reference. The 1^{th}, 2^{th}, 3^{th}, 5^{th}, 6^{th}, 7^{th} minutiae needs to be found in the query to generate two points in the protected query template to match PTR_1 and PTR_2. But in the query, the 7^{th} minutiae is not detected, thus PTQ_1 and PTQ_2 can not match PTR_1 and PTR_2.

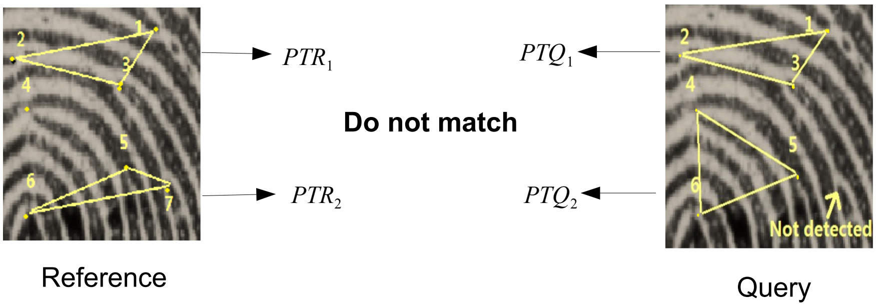


Figure 25: An example of failure to match

10 Conclusion and Contributions

This master thesis project proposed an interoperable minutiae template protection method named RIME. This method applies to the minutiae templates that are compliant to the ISO standard minutiae format. It derives features from minutiae triplets and transfers the features by using random projection. The accuracy of RIME using a number of different choices of features and parameter settings was tested. The lowest EER we obtained in the experiments is 16.11%. The methods for analyzing the irreversibility and unlinkability of RIME are provided.

This master thesis project contributed a fingerprint template protection method that enables interoperability among fingerprint authentication/verification systems that use ISO minutiae template format. It also provided an idea to derive local features from a minutiae template and an idea to achieve interoperability. It is a good reference for the researches based on the similar ideas.

11 Future work

The accuracy and privacy protection performance of RIME has been analyzed. The accuracy of RIME is expected to be increased in the future work. There are some possible improvements that can be tried:

1. Use ridge count between two minutia as additional information. Except for the minutiae location and orientation, ridge count between two minutia is also provided in the ISO minutiae format. We can try to include this information in the template protection method.
2. Look for better feature extraction method to extract features that are more reliable and have higher entropy.
3. Apply minutiae template level fusion to generate the reference to increase the reliability and entropy of the reference.

Bibliography

[1] Prabhakar, S., Pankanti, S., & Jain, A. 2003. Biometric recognition: Security and privacy concerns. *Security & Privacy, IEEE*, 1(2), 33–42.

[2] Maltoni, D., Maio, D., Jain, A., & Prabhakar, S. 2009. *Handbook of fingerprint recognition*. Springer-Verlag New York Inc.

[3] Jain, A., Nandakumar, K., & Nagar, A. 2008. Biometric template security. *EURASIP Journal on Advances in Signal Processing*, 2008, 113.

[4] ISO/IEC 19794-2:2005. *Information technology – biometric data interchange formats part 2: Finger minutiae data*. ISO, Geneva, Switzerland.

[5] ISO/IEC 19092:2008. *Financial services - Biometrics - Security framework.* ISO, Geneva, Switzerland.

[6] ISO/IEC 19794. *Information technology - biometric data interchange formats.* ISO, Geneva, Switzerland.

[7] Moore, S. Latest tests of biometrics systems shows wide range of abilities.

[8] Breebaart, J., Busch, C., Grave, J., & Kindt, E. 2008. A reference architecture for biometric template protection based on pseudo identities. *Proc. BIOSIG*, 137, 25–38.

[9] Gafurov, D., Bours, P., Yang, B., & Busch, C. 2010. Guc100 multi-scanner fingerprint database for in-house (semi-public) performance and interoperability evaluation. In *Computational Science and Its Applications (ICCSA), 2010 International Conference on*, 303–306. IEEE.

[10] 2011. Information Technology — Biometric data interchange Formats.

[11] 2382-37, I. D. *Information technology - Vocabulary - Part 37: Harmonized Biometric Vocabulary, 2011.* ISO, Geneva, Switzerland.

[12] Ratha, N., Connell, J., & Bolle, R. 2001. Enhancing security and privacy in biometrics-based authentication systems. *IBM systems journal*, 40(3), 614–634.

[13] Buchmann, J. 2004. Cryptographic hash functions. *Introduction to Cryptography*, 235–248.

[14] Simoens, K., Yang, B., Zhou, X., Beato, F., Busch, C., Newton, E., & Preneel, B. 2012. Criteria towards metrics for benchmarking template protection algorithms. In *2012 5th IAPR International Conference on Biometrics*.

[15] 2011. Information technology - Security techniques - Biometric information protection.

[16] Nagar, A., Nandakumar, K., & Jain, A. 2010. Biometric template transformation: a security analysis. In *Proceedings of SPIE*, volume 7541, 75410O.

[17] Jain, A., Ross, A., & Uludag, U. 2005. Biometric template security: Challenges and solutions. In *Proceedings of European Signal Processing Conference (EUSIPCO)*, 469–472.

[18] Teoh, A., Goh, A., & Ngo, D. 2006. Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12), 1892–1901.

[19] Teoh, A. & Ngo, D. 2005. Cancellable biometerics featuring with tokenised random number. *Pattern Recognition Letters*, 26(10), 1454–1460.

[20] Teoh, A., Ngo, D., & Goh, A. 2004. Personalised cryptographic key generation based on facehashing. *Computers & Security*, 23(7), 606–614.

[21] Le, A. & Picone, D. 1998. Linear discriminant analysis.

[22] Wold, S., Esbensen, K., & Geladi, P. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1), 37–52.

[23] Ratha, N., Chikkerur, S., Connell, J., & Bolle, R. 2007. Generating cancelable fingerprint templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(4), 561–572.

[24] Ratha, N., Connell, J., Bolle, R., & Chikkerur, S. 2006. Cancelable biometrics: A case study in fingerprints. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, 370–373. IEEE.

[25] Bazen, A. & Veldhuis, R. 2004. Likelihood-ratio-based biometric verification. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1), 86–94.

[26] Juels, A. & Sudan, M. 2006. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38(2), 237–257.

[27] Juels, A. & Wattenberg, M. 1999. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security*, 28–36. ACM.

[28] Moon, D., Lee, S., Jung, S., Chung, Y., Park, M., & Yi, O. 2007. Fingerprint template protection using fuzzy vault. *Computational Science and Its Applications–ICCSA 2007*, 1141–1151.

[29] Yang, S. & Verbauwhede, I. 2005. Automatic secure fingerprint verification system based on fuzzy vault scheme. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 5, v–609. IEEE.

[30] Uludag, U. & Jain, A. 2006. Securing fingerprint template: Fuzzy vault with helper data. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, 163–163. IEEE.

[31] Feng, Y. & Yuen, P. 2006. Protecting face biometric data on smartcard with reed-solomon code. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, 29–29. IEEE.

[32] Lee, Y., Bae, K., Lee, S., Park, K., & Kim, J. 2007. Biometric key binding: Fuzzy vault based on iris images. *Advances in Biometrics*, 800–808.

[33] Jain, A. & Li, S. 2009. Encyclopedia of biometrics.

[34] Hao, F., Anderson, R., & Daugman, J. 2006. Combining crypto with biometrics effectively. *Computers, IEEE Transactions on*, 55(9), 1081–1088.

[35] Dodis, Y., Reyzin, L., & Smith, A. 2004. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Advances in cryptology-Eurocrypt 2004*, 523–540. Springer.

[36] Li, Q., Guo, M., & Chang, E. 2008. Fuzzy extractors for asymmetric biometric representations. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, 1–6. IEEE.

[37] Arakala, A., Jeffers, J., & Horadam, K. 2007. Fuzzy extractors for minutiae-based fingerprint authentication. *Advances in Biometrics*, 760–769.

[38] Chang, E. & Roy, S. 2007. Robust extraction of secret bits from minutiae. *Advances in Biometrics*, 750–759.

[39] Zhou, X. 2007. Template protection and its implementation in 3d face recognition systems. In *Proceedings of SPIE Conference on Biometric Technology for Human Identification*, volume 6539, 214–225.

[40] Johnson, R. & Wichern, D. 2002. *Applied multivariate statistical analysis*, volume 4. Prentice Hall Upper Saddle River, NJ.

[41] Cloud, G. 1998. *Optical methods of engineering analysis*. Cambridge Univ Pr.

[42] Leedy, P. & Ormrod, J. 2005. *Practical research: Planning and design*. Prentice Hall Upper Saddle River, NJ.

[43] Medina-Pérez, M., García-Borroto, M., Gutierrez-Rodríguez, A., & Altamirano-Robles, L. 2012. Improving fingerprint verification using minutiae triplets. *Sensors*, 12(3), 3418–3437.

[44] Fingerprint verification competition. `http://bias.csr.unibo.it/fvc2002/`, 25.05.2012.

[45] Neurotechnology. `http://www.neurotechnology.com/`, 25.02.2012.

[46] Rodgers, J. & Nicewander, W. 1988. Thirteen ways to look at the correlation coefficient. *American Statistician*, 59–66.

[47] Nist fingerprint image software (nfis). http://fingerprint.nist.gov/NFIS/, 25.02.2012.

[48] Coffin, D. 2011. Two-factor authentication. *Expert Oracle and Java Security*, 177–208.

A RIME Matlab Scripts

```
function [EncMinutiae,EncMLength] = rime(Minutiae,Par,Key)
% Input:  Minutiae: original minutiae template from finger $f$
%         Par: parameters used in RIME
%              Par.Radius: r
%              Par.Normalization: the first row of Par.Normalization is the mean
%                                 the second row of Par.Normalization is the std
%              Par.DistTH
%              Par.Coefficient: Coefficient
%         Key: the transformation function for $f$
% Output: EncMinutiae: the protected template of Minutiae
%         EncMLength:  the number of points in the protected template EncMinutiae
mx = Minutiae.x;
my = Minutiae.y;
mtheta = Minutiae.theta;

EncMinutiae.x = [];
EncMinutiae.y = [];
EncMinutiae.theta = [];

OriMLength = length(mx);
if OriMLength < 4
   EncMinutiae.x = [];
   EncMinutiae.y = [];
   EncMinutiae.theta = [];
else

   m = 0;

    for i=1:OriMLength
        Cx(1) = mx(i);
        Cy(1) = my(i);
        Ctheta(1) = mtheta(i);

        d = (Par.Radius-sqrt((mx-Cx(1)).^2+(my-Cy(1)).^2)).^2;
        [d_ss,idxx] = sort(d,'ascend');
        idx_temp=idxx;
        idx_temp( find(idxx == i))=[];
        idx=[i,idx_temp];

        m = m + 1;
        Cx(2:3) = mx(idx(2:3));
        Cy(2:3) = my(idx(2:3));
```

```
Ctheta(2:3) = mtheta(idx(2:3));

b=GetPar(Cx,Cy,Ctheta,Par);

[Ptx,Pty,Pttheta]=transformation(b,Par,Key);

EncMinutiae.x = [EncMinutiae.x Ptx];
EncMinutiae.y = [EncMinutiae.y Pty];
EncMinutiae.theta = [EncMinutiae.theta Pttheta];

if d_s(3)^0.5-d_s(2)^0.5 < Par.DistTH

   m = m + 1;

   Cx(2) = mx(idx(3));
   Cy(2) = my(idx(3));
   Ctheta(2) = mtheta(idx(3));
   Cx(3) = mx(idx(2));
   Cy(3) = my(idx(2));
   Ctheta(3) = mtheta(idx(2));

   b=GetPar(Cx,Cy,Ctheta,Par);

   [Ptx,Pty,Pttheta]=transformation(b,Par,Key);

   EncMinutiae.x = [EncMinutiae.x Ptx];
   EncMinutiae.y = [EncMinutiae.y Pty];
   EncMinutiae.theta = [EncMinutiae.theta Pttheta];
end

if d_s(4)^0.5-d_s(3)^0.5 < Par.DistTH

    m = m + 1;

    Cx(2) = mx(idx(2));
    Cy(2) = my(idx(2));
    Ctheta(2) = mtheta(idx(2));
    Cx(3) = mx(idx(4));
    Cy(3) = my(idx(4));
    Ctheta(3) = mtheta(idx(4));

    b=GetPar(Cx,Cy,Ctheta,Par);

    [Ptx,Pty,Pttheta]=transformation(b,Par,Key);

    EncMinutiae.x = [EncMinutiae.x Ptx];
    EncMinutiae.y = [EncMinutiae.y Pty];
    EncMinutiae.theta = [EncMinutiae.theta Pttheta];
end
```

```
    end

    EncMLength=m;

end

function [b]=GetPar(Cx,Cy,Ctheta,Par)
% Parameter generation function.
% Input: Cx:     the x values of the triplet
%        Cy:     the y values of the triplet
%        Ctheta: the angle values of the triplet
%        Par:    the parameters used in RIME
% Output: b: the transformation parameter
        Cxx(1)=Cx(2)-Cx(1);
        Cxx(2)=Cx(3)-Cx(1);
        Cyy(1)=Cy(2)-Cy(1);
        Cyy(2)=Cy(3)-Cy(1);

        Cxx(3)=Cx(3)-Cx(2);
        Cyy(3)=Cy(3)-Cy(2);

        A=[Cxx.^2;
            Cyy.^2];

        l1 = sqrt(Cxx(1)^2+Cyy(1)^2);
        l2 = sqrt(Cxx(2)^2+Cyy(2)^2);
        l3 = sqrt(Cxx(3)^2+Cyy(3)^2);
        avel=sum(sqrt(sum(A,1)))/3;

        Cthetaa(1:2) = mod(Ctheta(2:3)-Ctheta(1),360);
        Cthetaa(3) = mod(Ctheta(3)-Ctheta(2),360);

        O1 = mod(atan2(Cy(2)-Cy(1),Cx(2)-Cx(1)),2*pi)*180/pi;
        O2 = mod(atan2(Cy(3)-Cy(2),Cx(3)-Cx(2)),2*pi)*180/pi;
        O3 = mod(atan2(Cy(1)-Cy(3),Cx(1)-Cx(3)),2*pi)*180/pi;

        theta1 = mod(Ctheta(1)-[O1,O3],360);
        theta2 = mod(Ctheta(2)-[O1,O2],360);
        theta3 = mod(Ctheta(3)-[O2,O3],360);

        beta12=Cthetaa(1);
        beta13=Cthetaa(2);
        beta23=Cthetaa(3);

        alpha11=theta1(1);
        alpha13=theta1(2);
        alpha21=theta2(1);
        alpha22=theta2(2);
        alpha32=theta3(1);
        alpha33=theta3(2);

        %l1, l2, l3, avel, beta12, beta13, beta23, alpha11, alpha13, alpha21, alph
        %features derived from a triplet. We choose k features from the 13 feature
        b = [alpha11, alpha22, alpha33];
        b = (b—Par.Normalization(1,:))./Par.Normalization(2,:);

function [Ptx,Pty,Pttheta]=transformation(b,Par,Key)
% Transformation function
% Input: b:   the transformation parameter
%        Par: the parameters used in RIME
%        Key: the transformation key
%Output: Ptx:     the x value of one point in the protected template
%        Pty:     the y value of one point in the protected template
%        Pttheta: the angle value of one point in the protected template
         v = matrix.m*b';

         Ptx = Par.Coefficient*v(1);
         Pty = Par.Coefficient*v(2);
         Pttheta = mod(Coefficient*v(3),360);
```