SECURITY/PRIVACY ANALYSIS OF BIOMETRIC HASHING AND TEMPLATE PROTECTION FOR FINGERPRINT MINUTIAE

by

Berkay Topçu

Submitted to
the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

SABANCI UNIVERSITY

June 2016

SECURITY/PRIVACY ANALYSIS OF BIOMETRIC HASHING AND TEMPLATE PROTECTION FOR FINGERPRINT MINUTIAE

APPROVED BY

Assoc. Prof. Dr. Hakan ERDOĞAN ....
(Thesis Supervisor)

Prof. Dr. Berrin YANIKOĞLU ....

Assoc. Prof. Dr. Müjdat ÇETİN ....

Assoc. Prof. Dr. Murat SARAÇLAR ....

Assoc. Prof. Dr. Olcay KURŞUN ....

DATE OF APPROVAL: ....

To my family. . . .

Acknowledgements

This Ph.D. experience is not only an immense source of pride for me but also a milestone in my life; one that will always recall some great moments and incredible people who I am honored to meet. I would like to express my deepest gratitude to all those people who accompanied me during all these years and make it so unforgettable.

First of all, I would like to sincerely acknowledge and express my gratitude to my advisor Dr. Hakan Erdoğan for his endless guidance in my professional growth and encouragement throughout my graduate studies. I enormously benefited from his wisdom, profound experience, far-sighted ideas, enthusiasm, and patience over these years. From the very first day of our collaboration, he always gave me the freedom to explore on my own and provided continuous support when I struggled. My knowledge in the field is deeply indebted to what he taught me, and it is my hope that this work offers something worthwhile in return. I am also thankful to him for setting high standards while teaching me how to do research, and for editing and commenting on revisions of my paper drafts.

I would like to extend my thanks to the remaining members of my committee, Dr. Müjdat Çetin, Dr. Berrin Yanıkoğlu, Dr. Murat Saraçlar and Dr. Olcay Kurşun, for giving generously of their time to read and comment on this manuscript. Dr. Yanıkoğlu also played a major role in this research with her unique blend of energy, professionalism, and knowledge. I am thankful for her great influence and support in the course of my studies. I am also grateful to my committee for their contributions and suggestions to the successful completion of this work. It has never been easy to answer their challenging questions, but they definitely helped me better understand the weaknesses and strengths of my research. I also would like to thank my professors from the Faculty of Engineering and Natural Sciences at Sabancı University for their great contributions to my undergraduate and graduate studies.

I am forever grateful to my dearest friends, Sezgin Akpınar, Emre Akşit, Kerem Başol, Mustafa Baytar, and Güvenir Kaan Esen for always being on my side and there for me when I need it. They have been a vital part of my life for such a long time; thus a life without their friendship is not a full and good one.

My special thanks go to one of the most valuable people in my life, Şeniz Demir, for her enormous support and guidance throughout the completion of this thesis. It has been great to feel her belief in me and my work, and to be able to reach her regardless of time and place.

I also would like to acknowledge the Scientific and Technological Research Council of Turkey (TUBİTAK) BİDEB for National Scholarship Programme (2211) for PhD students. Additionally, I would like to thank to my unit head Dr. Oktay Adalıer, and project manager and colleague Dr. Çağatay Karabat from TUBİTAK BİLGEM for their understanding during heavy periods. In addition, I would like to thank Melis Özgür Çetinkaya Demir, Edona Fasllija, Elif Üstündağ Soykan, and Muhammet Yıldız for creating a friendly working environment in the office.

Last and foremost, I would like to thank my family, Yılmaz, Reyhan, and Esra Topçu, for their unconditional love and continuous support throughout my entire life. I owe a great debt to them for trusting in me every time I have a dream and a goal to pursue.

SECURITY/PRIVACY ANALYSIS OF BIOMETRIC HASHING AND TEMPLATE PROTECTION FOR FINGERPRINT MINUTIAE

BERKAY TOPÇU

EE, Ph.D. Thesis, 2016

Thesis Supervisor: Hakan Erdoğan

Keywords: Biometrics, biometric template protection, face verification, fingerprint verification, biohashing.

Abstract

This thesis has two main parts. The first part deals with security and privacy analysis of biometric hashing. The second part introduces a method for fixed-length feature vector extraction and hash generation from fingerprint minutiae.

The upsurge of interest in biometric systems has led to development of biometric template protection methods in order to overcome security and privacy problems. Biometric hashing produces a secure binary template by combining a personal secret key and the biometric of a person, which leads to a two factor authentication method. This dissertation analyzes biometric hashing both from a theoretical point of view and in regards to its practical application. For theoretical evaluation of biohashes, a systematic approach which uses estimated entropy based on degree of freedom of a binomial distribution is outlined. In addition, novel practical security and privacy attacks against face image hashing are presented to quantify additional protection provided by biometrics in cases where the secret key is compromised (i.e., the attacker is assumed to know the user's secret key). Two of these attacks are based on sparse signal recovery techniques using one-bit compressed sensing in addition to two other minimum-norm solution based attacks. A rainbow attack based on a large database of faces is also introduced. The results show that biometric templates would be in serious danger of being exposed when the secret key is known by an attacker, and the system would be under a serious threat as well.

Due to its distinctiveness and performance, fingerprint is preferred among various biometric modalities in many settings. Most fingerprint recognition systems use minutiae information, which is an unordered collection of minutiae locations and orientations.

Some advanced template protection algorithms (such as fuzzy commitment and other modern cryptographic alternatives) require a fixed-length binary template. However, such a template protection method is not directly applicable to fingerprint minutiae representation which by its nature is of variable size. This dissertation introduces a novel and empirically validated framework that represents a minutiae set with a rotation invariant fixed-length vector and hence enables using biometric template protection methods for fingerprint recognition without significant loss in verification performance. The introduced framework is based on using local representations around each minutia as observations modeled by a Gaussian mixture model called a universal background model (UBM). For each fingerprint, we extract a fixed length super-vector of first order statistics through alignment with the UBM. These super-vectors are then used for learning linear support vector machine (SVM) models per person for verification. In addition, the fixed-length vector and the linear SVM model are both converted into binary hashes and the matching process is reduced to calculating the Hamming distance between them so that modern cryptographic alternatives based on homomorphic encryption can be applied for minutiae template protection.

BİYOMETRİK KIYIM İÇİN GÜVENLİK/MAHREMİYET ANALİZİ VE PARMAK İZİ OLAY NOKTALARI İÇİN ŞABLON KORUMA

BERKAY TOPÇU

EE, Doktora Tezi, 2016

Tez Danışmanı: Hakan Erdoğan

Anahtar Kelimeler: Biyometrik, biyometrik şablon koruma, yüz tanıma, parmak izi doğrulama, biyometrik kıyım.

Özet

Bu tez çalışması iki ana parçadan oluşmaktadır. İlk kısım biyometrik kıyım (hash) yönteminin güvenliğini ve mahremiyetini ele almaktadır. İkinci kısım ise parmak izi olay noktaları için sabit uzunlukta bir vektör ve kıyım oluşturma yöntemi sunmaktadır.

Biyometrik sistemlere hızla artan ilgi, güvenlik ve mahremiyet problemlerini arttrm ve dolayısıyla biyometrik şablon koruma yöntemlerinin geliştirilmesini de beraberinde getirmiştir. Biyometrik kıyım, kişinin biyometrisi ile kişisel bir gizli anahtarı birleştirerek güvenli bir ikili (binary) şablon oluşturur ve iki unsurlu bir biyometrik doğrulama yöntemi sunar. Bu tez çalışması, biyometrik kıyım yöntemini hem teorik açıdan hem de pratik uygulama yönünden analiz etmektedir. Biyometrik kıyımın teorik değerlendirmesi kapsamında binomial dağılımın serbestlik derecesine dayalı entropi kestirimini kullanan sistematik bir yöntem anlatılmaktadır. Buna ek olarak, yüz imgesi kıyımına yönelik özgün güvenlik ve mahremiyet atakları sunulmaktadır. Bu ataklar ile kişinin gizli anahtarının art niyetli bir saldırganca bilindiği durumlarda biyometrik tarafından sağlanan ilave koruma miktarı ölçülmektedir. Bu ataklardan ikisi bir-bit sıkıştırmalı algılama kullanan seyrek işaret geri kazanımına dayanmaktadır. Diğer iki atak ise en küçük işaret boyu çözümlerine dayanmaktadır. Bunlara ek olarak büyük bir yüz veritabanına dayalı gökkuşağı atağı da sunulmaktadır. Sonuçlar göstermektedir ki, kişisel anahtarın saldırgan tarafından bilindiği durumda biyometrik şablon açığa çıkma tehlikesi ile karşıya karşıya kalmakta ve aynı zamanda sistem de ciddi tehdit altında bulunmaktadır.

Parmak izi, yüksek ayırdediciliği ve başarımı dolayısıyla pek çok farklı biyometrik özellik arasından tercih edilmektedir. Parmak izi tanıma sistemlerinin tamamına yakını sıralı olmayan olay noktalarının konum ve yön bilgilerini kullanmaktadır. Fuzzy commitment ve diğer modern kriptografik alternatifler gibi ileri biyometrik şablon koruma yöntemleri sabit uzunlukta bir öznitelik vektörüne ihtiyaç duymaktadır. Dolayısıyla, bu yöntemler doğası gereği farklı sayıda olan parmak izi olay noktalarını korumak için kullanılamamaktadır. Bu tez çalışması, parmak izi olay noktaları kümesini dönmelere değişimsiz ve sabit uzunlukta bir vektör olarak ifade eden, özgün ve geçerliliği deneysel olarak gösterilmiş bir yöntem sunmaktadır. Bu sayede biyometrik şablon koruma yöntemlerinin ciddi bir performans kaybı olmadan parmak izi tanıma için kullanılabilmesi sağlanmıştır. Sunulan yöntem, her bir olay noktası etrafındaki yerel gösterimleri evrensel arka plan modeli (UBM) olarak adlandırılan bir Gaussian karışım modeli ile modellenen gözlemler olarak kullanmaktadır. Her bir parmak izi için, UBM ile olan doğrultusuna göre birinci dereceden istatistiklerin bir süper-vektörünü oluşturulmakta ve bu süper-vektörler, doğrulama işleminde kullanılmak üzere her bir kişinin doğrusal karar destek makinesi (SVM) modelini öğrenmek için kullanılmaktadır. Ayrıca, hem sabit uzunluktaki süper-vektör hem de doğrusal SVM modeli ikili bir kıyıma dönüştürülmüş ve karşılaştırma işlemi bu ikisi arasındaki Hamming uzaklığının hesaplanmasına indirgenmiştir. Böylelikle, parmak izi olay noktaları homomorfik (benzer yapılı) şifreleme temelli kriptografik alternatifler ile korunabilir hale gelmiştir.

Contents

List of Figures

List of Tables

Abbreviations

AAM	**A**dvanced **A**ttack **M**odel
ALT	**A**ttack in the **L**ong **T**erm
AFIS	**A**utomated **F**ingerprint **I**dentification **S**ystem
ALSH	**A**symmetric **L**ocality **S**ensitive **H**ashing
BIHT	**B**inary **I**terative **H**ard **T**hresholding
CM	**C**ollusion **M**odel
DCT	**D**iscrete **C**osine **T**ransform
DET	**D**ecision **E**rror **T**radeoff
EER	**E**qual **E**rror **R**ate
EM	**E**xpectation **M**aximization
FAR	**F**alse **A**cceptance **R**ate
FMT	**F**ourier **M**ellin **T**ransform
FRR	**F**alse **R**ejection **R**ate
FTA	**F**ailure **T**o **A**cquire
FTC	**F**ailure **T**o **C**apture
FTD	**F**ailure **T**o **D**etect
FTE	**F**ailure **T**o **E**nroll
FTP	**F**ailure **T**o **P**rocess
FVC	**F**ingerprint **V**erification **C**ompetition
FpVTE	**F**inger**p**rint **V**endor **T**echnology **C**ompetition
GMM	**G**aussian **M**ixture **M**odel
IAFIS	**I**ntegrated **A**utomated **F**ingerprint **I**dentification **S**ystem
ID	**I**dentity **D**ocument
IHT	**I**terative **H**ard **T**hresholding
IRIS	**I**nversion for the **S**ame **B**iometric **S**ystem

ISO	**I**nternational **O**rganization for **S**tandardization
JPEG	**J**oint **P**hotographic **E**xperts **G**roup
LBP	**L**ocal **B**inary **P**atterns
LDA	**L**inear **D**iscriminant **A**nalysis
LP	**L**inear **P**rogramming
LSH	**L**ocality **S**ensitive **H**ashing
MAP	**M**aximum **A** **P**osteriori
MCC	**M**inutiae **C**ylinder **C**ode
MLP	**M**ulti **L**ayer **P**erceptron
PCA	**P**rincipal **C**component **A**nalysis
PDF	**P**robability **D**istribution **F**unction
PIN	**P**ersonal **I**dentification **N**umber
PRNG	**P**seudo **R**andom **N**umber **G**enerator
RP	**R**andom **P**rojection
SAKC	**S**ecurity **A**fter **K**ey **C**hange
SDK	**S**oftware **D**evelopment **K**it
SMC	**C**omplex **S**pectral **M**inutiae
SML	**L**ocation based **S**pectral **M**inutiae
SMO	**O**rientation based **S**pectral **M**inutiae
SRP	**S**igned **R**andom **P**rojection
SVM	**S**upport **V**ector **M**achine
UBM	**U**niversal **B**ackground **M**odel
USB	**U**niversal **S**erial **B**us

Chapter 1

Introduction

1.1 Biometric Template Protection

Biometric traits (such as fingerprint, face, and iris) are inalienable and distinctive attributes that can be used in establishing personal identities. For instance, fingerprints are ubiquitous in that each and every person but those with some kinds of physical disabilities has fingerprints. Additionally, fingerprints are unique to each person and no more than one person has the same fingerprint. Distinguishing and to some extent permanent characteristics of biometric traits offer greater security and convenience than traditional forms of verification that are based on passwords or tokens (such as PIN numbers and ID cards). Biometric authentication systems have been used to authenticate personal identities in many real world applications such as electronic identity cards, border control systems with electronic travel documents, electronic payment systems, and forensics applications since they provide a fast, reliable, and secure electronic authentication mechanism. The societal importance of biometrics and its main contributions to our daily lives are enormous as succinctly stated in [2]:

> *Biometrics is not only a fascinating pattern recognition research problem but, if carefully used, is an enabling technology with the potential to make our society safer, reduce fraud and provide user convenience.*

Automatically determining the validity of an identity claim by a person is a critical task, but unfortunately, the knowledge-based mechanisms and similarly token-based authentication systems are not able to meet this challenge. Neither a token nor a password, which can be stolen or handed over easily, provides a unique link between a person and his identity. At the governmental level, e-Passports store fingerprints and face photos in Europe. For visa application and border control, the US visit program keeps records of 10 fingers and face images of each person. In addition, automated fingerprint identification systems (AFIS), which are fingerprint and criminal history systems, help local, state, and federal partners solve and prevent crime by catching criminals and terrorists with the use of automated fingerprint and latent search capabilities. FBI IAFIS includes not only fingerprints but also additional biometrics such as corresponding mug shots and photos of scars and tattoos.

However, widespread deployment of biometric authentication systems in real world applications brings about severe security and privacy concerns [3–5]. This is the main driving force behind significant research efforts put forward to protect biometric templates of users. In the literature, several biometric template protection methods have been proposed (e.g., fuzzy commitment scheme [6] and biohashing [7]) in order to overcome these concerns by securing biometric templates. As another advantage, protected templates ideally enable multiple secure references to be created from the same biometric data. These secure references are supposed to be unlinkable and non-invertible in order to achieve the desired level of security and to fulfill privacy requirements.

The main goals of template protection are i) security, ii) privacy protection ability, iii) and unlinkability [8]. *Security* of a protected template corresponds to the difficulty of creating a "pre-image" of the template that gives a positive authentication result. *Privacy protection ability* of a protected template involves irreversibility and privacy leakage. Irreversibility indicates the hardness in retrieving original biometric data and privacy leakage shows the amount of information exposed in protected templates about the biometric data [3]. Another motivation for template protection is to prevent *linking* protected templates. It should not be easy for an adversary to decide whether two protected templates belong to same subject or not (cross matching). Moreover, the combination of two or more protected templates should not reveal secrets or biometric features (leakage amplification).

Biometric hashing (biohashing) scheme is a transformation-based template protection method that projects an input biometric trait to a pseudo-random space. After a thresholding step, the biometric sample of a user is converted into a binary vector. Biohashing is used to secure different biometric modalities such as fingerprints [7], faces [9], and palms [10]. It uses a user specific secret key for creating a random projection specific to each user. The ability to revoke the biohash of a user by simply assigning a new secret key in cases where the secret key of a user is compromised is a major advantage of biohashing. It is also possible to generate different biohashes with different secret keys. This allows a person to enroll to different services using his unique biometric data and prevents linkability.

Due to increased inter-class variation and preserved intra-class variation, biohashing significantly improves the matching performance. On the other hand, this performance degrades if the secret key of the user is known to the adversary. However, empirical studies showed that even in such cases, the matching accuracy is still comparable to that of unprotected biometric templates.

Although biohashing methods have become very popular due to their high authentication performance and easy deployment into match-on-card applications, research recently showed that they may suffer from serious security and privacy problems [8, 11–13]. A comprehensive security and privacy evaluation of biometric template protection methods can be carried out by theoretically analyzing the underlying methodology and assessing its vulnerabilities under practical attacks. In this dissertation, we present the first successful theoretical evaluation of biometric hashing as required for thorough analysis, where the unpredictability of biohashes generated by random projection (RP) based biohashing scheme is quantified via estimated entropy. The amount of information a biohash carries is quantitatively analyzed by measuring the entropy of a biohash obtained from a face image. Furthermore, to assess to what extent a biohash is unpredictable once the secret key of a user is stolen, we calculate the entropy of biohashes obtained using the same key but using biometric data from arbitrary people.

From a practical point of view, the strength of transformation-based methods is based on the hardness of invertibility of the underlying transformation. Introduction of practical attacks against biometric template protection methods are interesting since they reveal vulnerabilities in these methods. If a practical attack can be found, then this simply shows that the method cannot be reliably used for template protection. In some studies [11, 14], computational inversion techniques for biohashing and practical security analysis of biohashes have been explored. In this work, we have also addressed the reconstruction of face recognition features from face biohashes with a novel use of two different sparse recovery techniques from one-bit compressed sensing measurements. In addition, we introduce two minimum-norm solution attacks and a rainbow attack which makes use of a large database of faces..

1.2 Template Protection for Fingerprint Minutiae

Among various biometric modalities, fingerprint is preferred in many settings, due to its distinctiveness and performance, as well as the practicality and low cost of fingerprint readers. Most fingerprint recognition systems depend on the comparison of minutiae which are the endpoints and bifurcations of fingerprint ridges. They are known to remain unchanged throughout an individual's lifetime and enable a very discriminative classification of fingerprints [2].

Increasing use of fingerprint identification as well as other biometric modalities raise privacy concerns significantly [15] and hence protecting biometric fingerprint templates (mostly minutiae templates) becomes a requirement. We need a fixed-length orientation-invariant fingerprint representation to be able to use advanced template protection algorithms such as fuzzy commitment and modern cryptographic alternatives based on homomorphic encryption. However, the number of minutiae in a fingerprint depends on various conditions. For instance, two impressions of the same finger might not have an equal number of minutiae due to difficulties in fingerprint imaging and automatic minutiae extraction. This difference may result from the placement of a finger on the fingerprint reader (rotation or translation), elasticity of the skin (non-linear distortion), dryness or wetness of the finger, or the current amount of pressure applied. In addition, in cases where two impressions of the same finger are captured by two different readers, differences in the sensing area and sensor intrinsic properties may lead to a varying number of minutiae.

Spectral minutiae representation [16] proposes a method for combining fingerprint recognition with template protection. It transforms a minutiae set into a fixed-length feature vector by representing minutiae as a magnitude spectrum. This transformation is invariant to translation. Furthermore, rotation and scaling become easily compensated translations under this transformation. In this work, we present the first successful implementation of biometric hashing for spectral minutiae.

In practice, an alignment based on singular points (core and delta) is required for spectral minutiae representation in order to achieve good recognition performances [17] because a large rotation or a translation might lead to partial overlap between different impressions of the same finger. Additionally, missing or spurious minutiae lead to lower matching performances. To overcome these drawbacks, we propose a novel framework that enables the generation of a fixed-length feature vector representation for fingerprint minutiae based on local representations unlike spectral minutiae.

In our new representation, each minutia is represented as a minutia patch which encodes its geometric relations with other closely located minutiae. A minutia patch is translated and rotated accordingly to eliminate the registration requirement due to the relative alignment of fingerprints. Thus, a rotation invariant representation is obtained. The distribution of minutiae patches is modeled via a single user-independent Gaussian mixture model (GMM) called universal background model (UBM) and a fingerprint is represented with its probabilistic alignment to the UBM mixture components. We obtain first-order statistics from the alignment to UBM mixture components and use them to form a super-vector to represent each fingerprint. We further train a linear SVM in this large-dimensional vector space to discriminate a person's fingerprint from other people's fingerprints. This idea is borrowed from speaker verification literature where each frame of an utterance is assumed as a separate observation and a similar GMM-SVM approach is used for verification [18]. In this approach each minutia patch is analogous to a frame of speech and a collection of minutia patches which forms a fingerprint is analogous to an utterance.

Even though the above approach obtains fixed length vectors for representing fingerprints and their linear SVM models are also vectors of the same size, the representations are not binary and they may not be directly used with template protection methods which require binary representations. Hence, we explore the use of asymmetric locality sensitive hashing (ALSH) to map these vectors into binary strings and the inner products between vectors are approximated by the Hamming distance between mapped binary strings. In this framework, both fingerprints and linear SVM models are represented as binary strings and the decision is made by thresholding the Hamming distance between them, but the mapping to binary domain is slightly different for fingerprint vectors and SVM models, hence the locality sensitive hashing is asymmetric. In this framework, a fixed-length minutiae vector is also transformed into a binary string using asymmetric locality sensitive hashing (ALSH) [19]. Our framework is able to create a fixed-length binary feature vector of fingerprints to represent minutiae information. This enables the protection of fingerprint minutiae via current template protection methods such as fuzzy commitment and biometric hashing as well as application of homomorphic encryption techniques.

1.3 Contributions

In this dissertation, biometric template protection methods are addressed. Biometric hashing is analyzed from security and privacy aspects. In addition, template protection for fingerprint minutiae is discussed in detail and novel solutions are proposed.

The contributions of this research are summarized as follows:

- This work presents the first successful theoretical evaluation of biometric hashing as required for thorough analysis where unpredictability of biohashes is quantified via estimated entropy.
- This work estimates entropy of biohashes using the degree of freedom of binomial distribution as described by Daugman [1]. Our work demonstrates that Daugman's entropy estimation is not restricted only to iris but can also be applied to other biometric modalities that can be represented with a fixed-length binary string and compared via Hamming distance.
- This work proposes four novel optimization-based methods that aim to reconstruct the feature vector from a biohash. Assuming that an adversary gains access to the biohash vector of a user and the corresponding secret key, these methods can be used to estimate a new real-valued feature vector from binary biohash and authenticate to the system.

- This work introduces the first practical security and privacy attacks against biohashes using one-bit compressive sensing framework. Apart from that, minimum norm solutions are discussed in detail and L_1 norm minimization is introduced in addition to the L_2 norm minimization which previously appeared in the literature. Finally, this work introduces a type of "rainbow attack" against biometric hashing systems.

- This work evaluates spectral minutiae representation in depth and proposes the first implementation of biometric hashing for spectral minutiae.

- This work describes an underlying framework that enables the generation of a novel fixed-length feature vector representation for fingerprint minutiae based on GMM-SVM approach. The framework allows biometric template protection methods to be applied to fingerprint minutiae.

- This work presents the use of asymmetric locality sensitive hashing for binary strings generation from GMM-SVM fingerprint features. This allows fast and efficient matching via Hamming distance.

1.4 Outline of the Dissertation

Chapter 2 discusses related work in various research areas that is relevant to our work.

Chapter 3 describes biometric hashing in detail and presents entropy analysis of biohashes.

Chapter 4 presents novel methods for reconstructing biometric features from biohashes via sparse recovery.

Chapter 5 presents spectral minutiae representation in detail and provides the first implementation of biometric hashing for spectral minutiae.

Chapter 6 describes an underlying framework that enables the generation of a novel fixed-length feature vector representation for fingerprint minutiae and presents a binary hash generation method.

Chapter 7 presents our conclusions and future plans on extending this research.

Chapter 2

Related Work

This chapter presents related work in several disparate fields that is relevant to our work and describes how our work both builds on and differs from this existing research. Section 2.1 presents the fundamentals of biometric recognition systems. Section 2.2 looks at research efforts aimed at enhancing security and privacy aspects of biometric recognition systems by protecting biometric templates of users. Section 2.3 discusses potential vulnerabilities of biometric template protection methods and possible attacks against biometric hashing. Section 2.4 discusses research efforts specific to protecting fingerprint minutiae templates. Section 2.5 discusses fixed-length minutiae representations that is required for template protection.

2.1 Biometric Recognition Systems

Biometric recognition (simply biometrics) refers to the use of distinctive physical/physiological (e.g., fingerprints, face, and iris) or behavioral (e.g., speech) characteristics for automatically recognizing the identity of an individual or verifying/authenticating his claimed identity. These characteristics are called as biometric identifiers or traits.

Recognizing a person by his body and linking it to an identity is a very powerful tool for identity management. Biometrics is becoming an essential component of effective person identification solutions since biometric identifiers cannot be shared or misplaced, and they intrinsically represent individuals' bodily identities.

Three main management tools for the identification of a person are: i) what you have (i.e., ID cards), ii) what you know (i.e., password or PIN), and iii) who you are (i.e., biometrics). Biometrics are accepted as more reliable in recognizing a person than traditional token or knowledge-based methods due to their inalienable nature (e.g., they cannot be easily misplaced, forged, or shared). Some biometric characteristics that have been used for automated recognition include fingerprints, iris, face, hand or finger geometry, retina, voice, signature, and keystroke dynamics.

Automated biometric recognition systems consists of the following steps. A biometric sample is taken from an individual, for instance, a fingerprint or an iris scan, which might be represented by an image. Representative data (a biometric template) are often extracted from that sample. This biometric data, either the image or the template or both, is then stored on a storage medium which could be a database or a distributed environment (e.g., smart cards). All these phases constitute the enrolment process.

At a later stage, if a person presents himself to the system, the system will ask the person to submit his biometric characteristic(s). The system will then compare the image of the submitted sample (or the template extracted from it) with the biometric data/template taken during enrolment. The person is then recognized and accepted by the system if a match is obtained. If there is no match, the person is not recognized and “rejected” by the system.

Depending on the application context, a biometric system may either perform the verification or identification task:

- A verification system authenticates a person’s identity by comparing the captured biometric characteristics with his previously captured biometric reference template that is pre-stored in the system. It conducts a one-to-one matching to confirm whether the claimed identity of the individual is true.
- An identification system recognizes an individual by searching the entire enrolment template database for a match by conducting one-to-many comparisons.

Although biometrics promise to correctly identify or validate the identity of a subject, in practice, a biometric system is a pattern recognition system that inevitably makes some incorrect decisions. Some of the main source of errors are capture systems (i.e., Failure to Detect (FTD) and Failure to Capture (FTC)) and feature extraction (i.e., Failure to Process (FTP)). These kinds of errors can be combined into a single measure which is called as the "Failure to Acquire (FTA)". Another source of errors, named as the "Failure to Enroll (FTE)", is observed when there is not enough discriminatory information present in the feature sets.

Throughout this work, we focus on the verification task where a one-to-one matching between a reference biometric template and a query biometric template is performed. Two types of errors that can be committed by a verification system are the "false match" and "false non-match". False match corresponds to mistaking templates from two different subjects as belonging to the same subject. False non-match corresponds to mistaking two templates of the same subject to be from two different subjects. Although they do not exactly stand for each other, false acceptance and false rejection are commonly used in the same context.

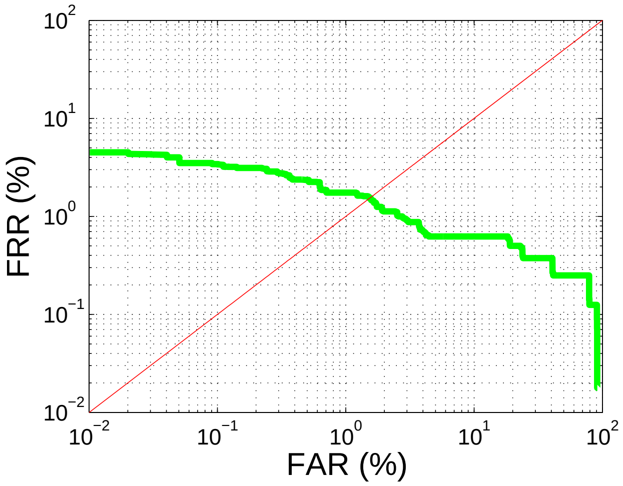


Figure 2.1: A sample ROC

In this work, we use the "False Acceptance Rare (FAR)" and "False Rejection Rate (FRR)" for evaluating the verification performance of biometric systems. There is a trade-off between these two types of errors since we can decrease one by increasing the other one. This is achieved by changing a decision threshold. We can plot FAR versus FRR in a detection error trade-off (DET) curve. An example DET curve is shown in Figure 2.1. Each point on the curve corresponds to using a different decision threshold. Same information can also be conveyed using a receiver operating characteristic (ROC) curve which plots true accept rates versus false reject rates. We also employ the "Equal

Error Rate (EER)" of a verification system, which is the error rate at a point where FAR and FRR are identical.

2.2 Biometric Template Protection Methods

Biometric recognition systems enable fast, reliable, and secure electronic authentication, however, their large scale deployment in real world applications causes privacy and security concerns [3–5]. Biometric systems are not foolproof and a critical vulnerability that is unique to biometrics systems is the possession of stored templates by adversaries [11]. Biometric data might reveal sensitive information such as race, gender, and certain medical conditions. Since biometric traits are supposed to be permanent and unique to an individual, stolen templates can be used as unique identifiers to link information across different applications. Moreover, biometric modalities are limited in number and they cannot be easily revoked to obtain another template as seen in the use of passwords. Therefore, it is essential to ensure the security of biometric templates and to protect biometric data. In the literature, several biometric template protection methods have been proposed [15] (e.g., fuzzy commitment scheme [6] and biohashing [7]) to overcome these concerns by securing biometric templates (e.g., face and fingerprint). Biometric template protection methods store a modified version of the biometric template and reveal as little information about the original biometric trait as possible without losing the capability to identify a person.

Template protection methods can be categorized into two groups: i) biometric cryptosystems [15] (e.g., fuzzy commitment [6], fuzzy vault [20]) and ii) transformation-based methods/salting [2] (e.g., biohashing [7]). Biometric cryptosystems either bind secrets into biometric data to form a secure biometric template or generate secrets from biometric data with the help of some auxiliary data. The secrets can be successfully retrieved during a genuine verification attempt. The helper or auxiliary data does not reveal significant information about the biometric or the key. On the other hand, transformation-based approaches distort or randomize biometric data with the use of non-invertible functions so that the original data cannot be reconstructed from transformed templates. Biometric templates are transformed using parameters derived from external information such as user keys or passwords.

Biohashing or biometric hashing [7, 9] is one of the transformation-based methods, in which the biometric template of the user is transformed into a protected binary string through multiplication with a pseudo-random projection matrix and quantization. Due to increased inter-class variation and preservation of intra-class variation, biohashing significantly improves verification accuracy when the secret key is kept secure and unknown to adversaries. In this thesis, we use the terms biohashing and biometric hashing synonymously, even though we think biometric hashing is a more descriptive name.

In addition to the increased performance of the protected templates when the secret key of a user is kept safe, another advantage of biometric hashing lies in the ease of revoking a transformed template by changing the associated secret key. Furthermore, using the same biometric data, a user can be authenticated to different services through different biohashes generated from distinct secret keys. This way, two records that are presented to two different systems cannot be linked and activities of the user is kept private.

2.3 Security and Privacy Evaluation of Biometric Hashing

Biometric hashing uses a unique secret key in order to randomize biometric template of each user. It is a two factor authentication system in which both the biometric modality and the secret key of a user have to be presented during authentication. Although biohashing methods have become very popular due to their high authentication performance and easy deployment into match-on-card applications, research recently showed that they might suffer from serious security and privacy problems [8, 11, 13, 21].

We believe that it is necessary to study the security and privacy preservation capabilities of biometric hashing especially when the secret key is compromised. If the key is always assumed to be kept secure, an authentication system which checks the accuracy of the entered key will achieve a zero verification error even without any need for biometric data.

2.3.1 Unpredictability of Biohashes

A comprehensive evaluation of biometric template protection methods can be carried out by theoretically analyzing the underlying methodology and assessing its vulnerabilities under practical attacks. For biometric cryptosystems, there exist some theoretical analyses utilizing information theoretical metrics (e.g., entropy, conditional entropy, and mutual information) or metrics used in cryptanalysis (e.g., min-entropy, average min-entropy, guessing entropy, and conditional guessing entropy) [8]. However, the applicability of these metrics to empirical evaluation and their computation in practice are still unknown and need further investigation. Unfortunately, transformation-based methods lack any such theoretical analysis.

In this work, we present the first successful theoretical evaluation of biometric hashing as required for thorough analysis where the unpredictability of biohashes generated by random projection (RP) based biohashing scheme is quantified via estimated entropy.

2.3.2 Irreversibility of Biohashes

The security performance of a biohashing scheme under the assumption of a known key is analyzed in [22] and [23], and biohashing is concluded to be a good biometric randomization algorithm with a high risk of compromising the biometric information. If the secret key of a user is compromised, the security of the protected template is at stake and it is only dependent on the non-invertibility of the biohash (i.e., it should be hard for an adversary to approximate the biometric feature vector from the biohash and the secret key). The reconstruction of a sufficiently similar feature vector that provides a close biohash to the original one, called a pre-image attack (masquerade attack), is a major threat to the template protection capability of a biometric hashing scheme. It is not sufficient to make a function "lossy" (not one-to-one) in order to have a one-way function [24]. The biohashing method of Ngo et al. is presented as a one-way function [9], however, we show that this is not the case (in the cryptographic sense) and biometric hashing is not pre-image attack resistant if the secret key that is used for generating a biohash is known to the adversary.

In the first study that investigates the invertibility of a biometric hashing algorithm [25], it was assumed that the biohash of a user and the corresponding random projection matrix are available to an adversary. Each dimension of the biohash vector was mapped to the set $\{-1, 1\}$ (by mapping [0]→[-1] and [1]→[1]) and the resulting vector was multiplied with the pseudo-inverse of the random projection matrix. A new biohash created from the estimated biometric feature vector was used to perform imposter attacks. A similar approach that uses the pseudo inverse of a random projection matrix was also presented in [26]. In [27], a new method was proposed to generate a biometric feature from biohashes using genetic algorithms. For each biohash in a database, the proposed genetic algorithm was applied to approximate the value of the biometric feature given the corresponding secret key.

A detailed analysis of irreversibility of biohashes was performed by Feng et al. [14] where the details of the random projection is solved using perceptron learning. It was assumed that the attacker does not have the secret key of the user and the parameters of the random projection are estimated using stolen biohashes and a local biometric database. The main difference of this study is that the method requires several stolen biohashes from several distinct subjects (68 subjects - 105 images/subject for one database and 350 subjects - 40 images/subject for another database) for parameter estimation. It was assumed that the whole system is available to the adversary as a black box and the matching scores could be eavesdropped. A local face dataset (3500 different local faces) was presented to the system along with a common token and every local binary template was matched against every stolen template. Using the matching scores and the stolen biohashes, local binary biohashes corresponding to the local face database were calculated, which were used for iterative perceptron learning to estimate the projection parameters. Once the parameters of the random projection were estimated, they could be used to generate synthetic real-valued features from a stolen biohash which is another perceptron problem.

In another recent study, Nagar et al. [11] presented a method to recover a close approximation to the original biometric features given the binary biohash vector of a subject and the transformation parameters by formulating the problem as an optimization problem. A database of unrelated biometric features was used for optimization. For each unrelated biometric feature vector from the database, a new feature vector was estimated by minimizing the Euclidean distance between the new feature vector and the unrelated biometric feature vector subject to the consistency criterion (i.e., the new biohash created from the estimated feature vector exactly matches the original biohash). The estimated feature vector was computed by taking the weighted average of t number of trials where the weight was the Hamming distance between the original biohash and the estimated one. This promising approach attempts to invert biohashes in a similar set-up with our proposed methods. Therefore, we compare our algorithms in terms of verification errors and computation times with this attack.

In this thesis, we propose four different novel optimization-based methods that aim to predict the feature vector and/or the biometric image itself. Here, we assume that an adversary gains access to the biohash vector of a valid system user and the corresponding secret key, and estimates a new real-valued feature vector from the binary biohash in order to authenticate to the system. Novel feature estimation methods are in the focus of this study.

Our novel contributions regarding the reversibility of biohashes can be stated as follows. Practical security and privacy attacks against biohashes using one-bit compressive sensing framework are introduced. Apart from that, minimum norm solutions are discussed in detail and L_1 norm minimization is introduced in addition to the L_2 norm minimization which appeared in the literature before. Finally, this study introduces a type of "rainbow attack" against biometric hashing systems. The differences between the existing attacks and our proposed attack are given in Table 2.1 in terms of assumptions and related security and privacy issues.

2.4 Template Protection for Fingerprint Minutiae

Template protection schemes require either a fixed length feature vector representation or a binarized string as input. Thus, a variable length minutiae representation of a fingerprint cannot be directly used in combination with these schemes. In addition, some template protection schemes designed specifically to work with unordered sets of varying number of minutiae (e.g., fuzzy vault [28]) experience degradation in matching accuracy due to alignment issues and nonlinear distortion [29].

Fuzzy vault scheme secures a set of r minutiae points by generating a uniformly random cryptographic key of L bits and transforming it into a polynomial P of degree k (where

TABLE 2.1: Existing biohash inversion attacks

Method	**Assumptions**	**Security**	**Privacy**
Multiply with the pseudo-inverse of the random projection matrix [25, 26]	- Random projection matrix is available - Threshold is fixed and it is 0 - Wavelet FMT face features	Attack with biohash from estimated features: - existing key - a new key is assigned and stolen again	
Genetic algorithms [27]	- Random projection matrix is available - Threshold is fixed and it is 0 - Fingercode features	1) Attack with biohash from estimated features: - existing key - a new key is assigned and stolen again 2) Average distance between real and approximated features	
Solve a constrained minimization of distance between estimated features and unrelated feature vector [11]	- Random projection matrix is available - Threshold is available - A database of unrelated features - Eigenface features	Attack with biohash from estimated features: - existing key - a new key is assigned and stolen again	Reconstructed face images from estimated vector using PCA inversion
Perceptron-learning with hill climbing & MLP modeling with customized hill-climbing [14]	- Several biohashes of various different subjects are available (other methods assume availability of a single stolen biohash) - Attacker can access the matching scores of the system - Secret key of the user is available	Identification scenario, where biohash generated from each synthetic face is matched against the stolen templates	Adversary has access to output of feature extractor given a face image & applies hill-climbing attack to generate synthetic face images
Methods **proposed** and discussed in this study: - **Sparse recovery** - Min-norm solutions	- Random projection matrix is available - Threshold is available - Eigenface features	1) Attack with biohash from estimated features: - existing key - a new key is assigned and is unknown - a new key is assigned and stolen again 2) Verification accuracy using the real features as gallery and approximated features as probe	Orthogonal linear face features (i.e., PCA and LDA): transformation matrix is known and its inverse is used to reconstruct face images

$k < r$). All the minutiae points in a fingerprint is then evaluated on this polynomial and the obtained set of points is secured by hiding them among a large set of randomly generated chaff points that do not lie on the polynomial P. The polynomial evaluation of the combination of genuine and chaff points constitute the vault. During authentication, the polynomial P can be successfully reconstructed by identifying the genuine points in the vault that are associated with the minutiae of the enrolled fingerprint if the query fingerprint is sufficiently close.

Attacks via record multiplicity, stolen key inversion attack and blended substitution attack are some specific attacks against a fuzzy vault [30]. If an attacker obtains two different vaults generated from the same biometric data, he can easily identify the genuine points and decode the vault. In addition, if an adversary learns the key embedded in the vault, he can decode the vault and obtain the biometric template. Furthermore, an adversary can substitute a few points in the vault using his fingerprint minutiae without being detected, since the vault contains a large number of chaff points. Thus, both the genuine user and the attacker can successfully authenticate to the system under the same identity (i.e., blended substitution [29]).

One of the earliest works on fingerprint template protection has secured minutiae information x, y, θ separately [31]. In a later study, FingerCode feature (a texture based fingerprint representation without minutiae information [32]) has been protected via biohashing [7]. Another branch of research has focused on securing each minutia separately. Yang et al. [12, 21] have proposed methods to extract a binary secure hash bit string from each minutia and its vicinity using minutiae information only. A more recent study similarly has used neighboring minutiae information along with texture information around each minutia and secured each minutia feature vector by biohashing [33]. Protected Minutiae Cylinder-Code (P-MCC) [34], one of the most accurate algorithms proposed recently, has secured each MCC structure that corresponds to a single minutia. All these studies have represented a single minutia with a fixed length binary string therefore matching between variable length final templates has been addressed as a minutiae pairing problem.

2.5 Fixed-length Feature Representation for Minutiae

Unfortunately, only a limited number of studies has presented methods for converting a minutiae set into fixed length feature vectors. In the work of Sutcu et al. [35], binary features were extracted by counting the number of minutiae present in randomly chosen cuboidal patches in the (x, y, θ) space occupied by the minutia. To chose a cuboid, an origin was selected uniformly at random in (x, y, θ) space, and the dimensions along the three axes were also randomly chosen. Next, the threshold was defined as the median of the number of minutiae points in the chosen cuboid, measured across the whole training set. The threshold value might differ for each cuboid based on its position and volume. If the number of minutiae points in a randomly generated cuboid exceeded the threshold, then a 1-bit was appended to the feature vector, otherwise a 0-bit was appended. N such random selections of cuboid resulted in an N-bit feature vector.

Nagar et al. [36] improved over [35] in a fundamental way such that each cuboid generates a richer feature set from which a larger number of bits could be extracted and those with the highest determinability are used for matching. Corresponding to each randomly chosen cuboid, they introduced three minutiae-based features: (i) *aggregate wall distance*: the summation of the closest distance of each minutia from the cuboid boundary, (ii) *minutiae average*: the average coordinate of all minutiae present in each cuboid in a given fingerprint sample, and (iii) *minutiae deviation*: the standard deviation of minutiae coordinates present in each cuboid in a given fingerprint sample. The extracted features were binarized using the median value of a given feature calculated over all enrolled fingerprints. Using the median value as threshold ensured that each bit has equal probability of being 1 or 0. The main limitation of this approach is that it requires the fingerprints to be aligned beforehand [37].

Bringer et al. [38] characterized a fingerprint in terms of its similarity to each representative local minutiae vicinities in a set of fixed size. This fixed size set was extracted from a representative database of all existing vicinities in the world of fingerprints. For a fingerprint, a feature vector that contains the similarities of its vicinities to those of the representative set was produced. The reported verification performance was far from the classical minutiae matching algorithms. This was attributed to purely local approach of the encoding algorithm since it deals well with local distortions of a fingerprint but lacks global coherency. In their follow up work [39], more discriminative information was added to distinguish impostors with high scores from genuine scores by using localization information of vicinities which increased the global coherency.

In the spectral minutiae representation [16, 17], each minutia location was coded by an isotropic two-dimensional Gaussian function in the spatial domain. Here, minutiae were represented as a magnitude spectrum and their orientations were incorporated by assigning each Gaussian a complex magnitude. Only the magnitude spectrum was considered and it was sampled on a log polar grid to obtain a fixed length vector. It is possible to perform matching between two spectral minutiae vectors without aligning them first since the magnitude spectrum is invariant to rotation and translation due to the shift, scale, and rotation properties of the Fourier transform. However, in practice, alignment based on singular points (core and delta) is required to achieve a good recognition performance [17] because a large rotation or translation may lead to partial overlap between different impressions of the same finger. It should be noted that spectral minutiae representation uses the global position and orientation information of the minutiae thus already include relations of minutiae to each other.

In our study, we evaluate spectral minutiae representation in depth and propose the first implementation of biometric hashing for spectral minutiae [40]. Next, we describe an underlying framework that enables the generation of a novel fixed-length feature vector representation for fingerprint minutiae. Also, a method based on asymmetric locality sensitive hashing is proposed to generate binary strings from fixed-length minutiae vectors.

Chapter 3

Biometric Hashing and Its Entropy

Biometric hashing is a vector based template protection method that is used to secure various biometric modalities such as fingerprint [7], face [9], palm [10], etc. In a typical biometric hashing scheme, the input biometric modality is represented as a vector of real numbers of length n, $\mathbf{x} \in \mathrm{R}^n$. After multiplying with a random matrix and applying a threshold, this representation is converted to a binary string.

Biometric hashing (simply biohashing) schemes are simple yet powerful biometric template protection methods [41–45]. Biohash is a binary and pseudo-random representation of a biometric template and biometric hashing schemes perform an automatic verification of a user based on his biohash (a binary string). Two inputs of a biometric hashing scheme are: i) biometric template and ii) user specific secret key. A biometric feature vector is transformed into another space using a pseudo-random set of vectors which are generated from the user's secret key. Then, the result is binarized to produce a pseudo-random bit-string which is called the biohash. The random projection matrix is unique and specific to each user and it can be stored in a USB token or a smartcard. In a practical system, a user specific random matrix is calculated using a seed (a user specific secret key) that is stored in a USB token or a smartcard microprocessor through a pseudo random number generator. The seed is the same with that used during the enrollment of a user and is different among different users and different applications [7]. This allows revocability of the subject's biohash in case it is compromised. Also, the same biometric trait of a subject can be used in different biometric recognition systems without constituting privacy threat as two biohashes of the same person with different keys are unlinkable.

In an ideal case, the distance between the biohashes belonging to biometric templates of the same user is expected to be relatively small. On the other hand, the distance between the biohashes belonging to different users is expected to be sufficiently high which enables higher recognition rates. The user is enrolled to the system at the enrollment stage. At the authentication stage, the user provides his biometric data and secret key to the system in order to prove his identity.

In the next section, we describe the random projection (RP) based biohashing scheme proposed by Ngo et al. [7] for face verification.

3.1 Enrollment Stage

The first stage in a biometric recognition system is the enrollment stage in which a user is introduced to the system for the first time. His biometric record is captured and converted to a reference biometric template which will be compared to a fresh sample at the authentication stage. This biometric template can be stored either in a central database or a smart card that will be in possession of the user.

3.1.1 Feature Extraction

At this phase, face images that are collected during the enrollment stage are used as the training set. The set has training face images belonging to registered users, $\mathbf{I}_{i,j} \in \mathrm{R}^{m \times n}$ where $i = 1, \ldots, K$ and K denotes the number of users, and $j = 1, \ldots, L$ and L denotes the number of training images per user. Each face image is represented as a vector, $\mathbf{y} \in \mathrm{R}^{(mn)\times 1}$. Then, the Principle Component Analysis (PCA) [46] is applied to face images in the training set for feature extraction:

$$\mathbf{x} = \mathbf{A}(\mathbf{y} - \mu), \tag{3.1}$$

where $\mathbf{A} \in \mathrm{R}^{k\times(mn)}$ is the PCA matrix trained by the face images in the training set, μ is the mean face vector, and $\mathbf{x} \in \mathrm{R}^{k\times 1}$ is the vector containing PCA coefficients ($k < mn$).

3.1.2 Random Projection

At this phase, a pseudo random projection (RP) matrix, $\mathbf{R} \in \mathrm{R}^{\ell \times k}$, is generated to transform the PCA coefficient vectors. The RP matrix elements are independent and identically distributed ($i.i.d$) and generated from a Gaussian distribution with zero mean and unit variance by using a Pseudo Random Number Generator (PRNG) with a seed derived from the user's secret key. The RP matrix projects the PCA coefficients onto an ℓ-dimensional space:

$$\mathbf{z} = \mathbf{R}\mathbf{x}, \tag{3.2}$$

where $\mathbf{z} \in \mathrm{R}^{\ell \times 1}$ is an intermediate biohash vector.

3.1.3 Quantization

At this phase, elements of the intermediate biohash vector $\mathbf{z}$ are binarized with respect to a threshold:

$$\mathbf{b}\left(k\right) = \begin{cases} 1, & \mathbf{z}\left(k\right) \geq \beta, \\ 0, & \text{otherwise}, \end{cases} \tag{3.3}$$

where $\mathbf{b} \in \ \{0,1\}^{\ell}$ denotes the biohash vector of the user and β denotes the quantization threshold which can be 0 (sign operator) or the mean value of the intermediate biohash vector $\mathbf{z}$, depending on the system design.

After enrollment, biometric hashes are stored in a database or in a smart card.

3.2 Authentication Stage

At the authentication stage of a biometric system, an identity claim of a user is evaluated and a decision (YES/NO) is given depending on the result of this evaluation. The fresh biometric sample of the claimer is matched against the enrollment record of the subject. Authentication result of the system depends on the similarity (or distance) between

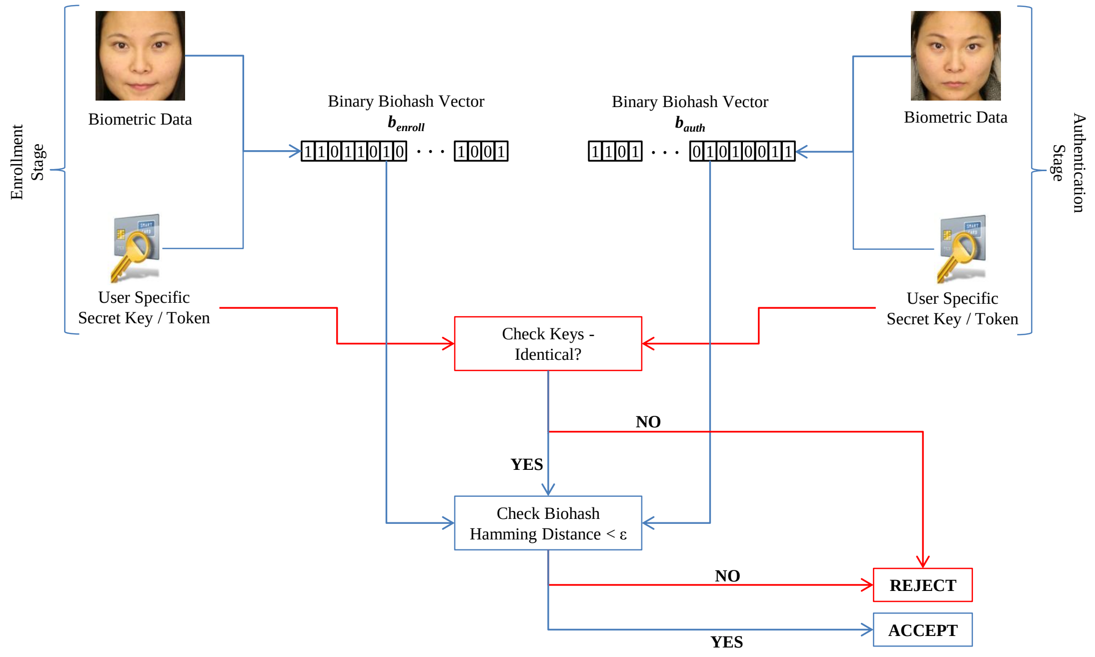


FIGURE 3.1: Biometric hashing verification setup

these two biometric templates. Throughout this thesis, authentication and verification are used interchangeably and both terms refer to a one-to-one matching.

At the authentication stage of the biometric hashing system, a claimer sends his face image $\tilde{\mathbf{I}} \in \mathrm{R}^{m \times n}$ and his secret key to the system. The system computes the claimer's test biometric hash vector by using the same procedures as in the enrollment phase. The user is authenticated when the Hamming distance between $\mathbf{b}_{enroll}$ (which denotes the biohash of the user generated at the enrollment stage) and $\mathbf{b}_{auth}$ (which denotes the biohash of the user generated at the authentication stage) is below a pre-determined distance threshold ϵ as follows:

$$\sum_{k=1}^{n} \mathbf{b}_{enroll}\left(k\right) \oplus \mathbf{b}_{auth}\left(k\right) \leq \epsilon \tag{3.4}$$

where $\oplus$ denotes the binary XOR (exclusive OR) operator. The system computes the Hamming distance between the test biometric hash vector and the claimed user's reference biometric hash vector stored in the database (or in the smart card). If the Hamming distance is below the pre-determined distance threshold, the claimer is accepted; otherwise, the claimer is rejected (Figure 3.1).

The remaining of this chapter presents the first successful theoretical evaluation of biometric hashing as required for thorough analysis where the unpredictability of biohashes generated by random projection (RP) based biohashing scheme is quantified via estimated entropy. Since a random projection and quantization method is required in our framework, the first study of Ngo et al. [9] among all other recent alternatives such as [47] was chosen since none has an effect on our entropy estimation method. The amount of information a biohash carries is quantitatively analyzed by measuring the entropy of a biohash obtained from a face image. Furthermore, to assess to what extent a biohash is unpredictable once the secret key of a user is stolen, the difference in the entropy of the original biohash and the entropy of the one created by using the stolen key along with the biometric feature of an arbitrary person is used.

We conduct experiments in a face verification set-up considering two different threat scenarios. Our results shows that the entropy of a biohash is almost equal to its bit length when the secret key of each user is kept safe. However, in the advanced threat scenario where the secret key of a user is compromised, the discriminative effect of the random projection is lost and the entropy of the biohash is limited to the entropy of the biometric feature. This is consistent with the study of Adler et al. [48] which shows that the biometric information for a person could be calculated by the relative entropy between the feature distributions of that person and the population (practically measured to be approximately 40 bits).

3.3 Entropy Prediction for Biohashing

The entropy of a random variable measures it uncertainty. In other words, it is a measure of the average amount of information required to describe a random variable. An important theoretical measure for biometric template protection methods is the entropy loss or mutual information (defined as the difference between unconditional and conditional entropies) [49]:

$$I(B;K) = H(B) - H(B|K), \tag{3.5}$$

where $H(B)$ is the entropy of biohash B and $H(B|K)$ is the conditional entropy of B where the corresponding secret key K is known (i.e., stolen by an adversary). In [15], the entropy of a biometric template is defined as the measure of the number of different identities that are distinguishable by a biometric system and it is a powerful indicator of its unpredictability. However, theoretical estimation methods are required to assess the entropy of a biohash since how to calculate that entropy is not immediately clear. One approach is to compute the bit-wise entropy of a biohash where the entropy of each bit location is calculated using a large database of biohashes [50]. Since this approach assumes that the bits of a biohash are independent and identically distributed, the predicted entropy is overestimated.

3.3.1 Daugman's Entropy Estimation

Daugman proposed a method for estimating the entropy of iris phase codes [1]. Iris phase codes, bit strings of length 2048, are compared using the normalized Hamming distance and the ratio of the number of disagreeing bits to the number of total bits are used to assess the degree of dissimilarity between two bit strings. A low dissimilarity ratio between two iris codes are accepted as belonging to the same eye whereas as from different eyes if it is close to 0.5.

Comparing bits corresponds to a Bernoulli trial and a binomial distribution is the distribution of the sum of n Bernoulli trials, each with the same probability. By observing the inter-class distance distribution over a large iris database, Daugman concluded that the distribution of the normalized Hamming distances between iris codes are normalized binomial with an observed mean of 0.499. Correlated Bernoulli trials reduce the effective number of trials but the output is still binomially distributed [51]. In iris phase codes, only a small number of bits are mutually independent, therefore the effective number of bits is not 2048 (number of bits in a phase code) but 249 and this corresponds to the entropy of an iris phase code [1]. In Figure 3.2, the observed distribution is plotted against the theoretical normalized binomial (the solid curve), which shows the close fit between them.

3.3.2 Entropy of Biometric Hashing

Biohashes are bit strings as iris codes and are compared via Hamming distance during authentication. In this work, we utilize these similarities between biohashes and

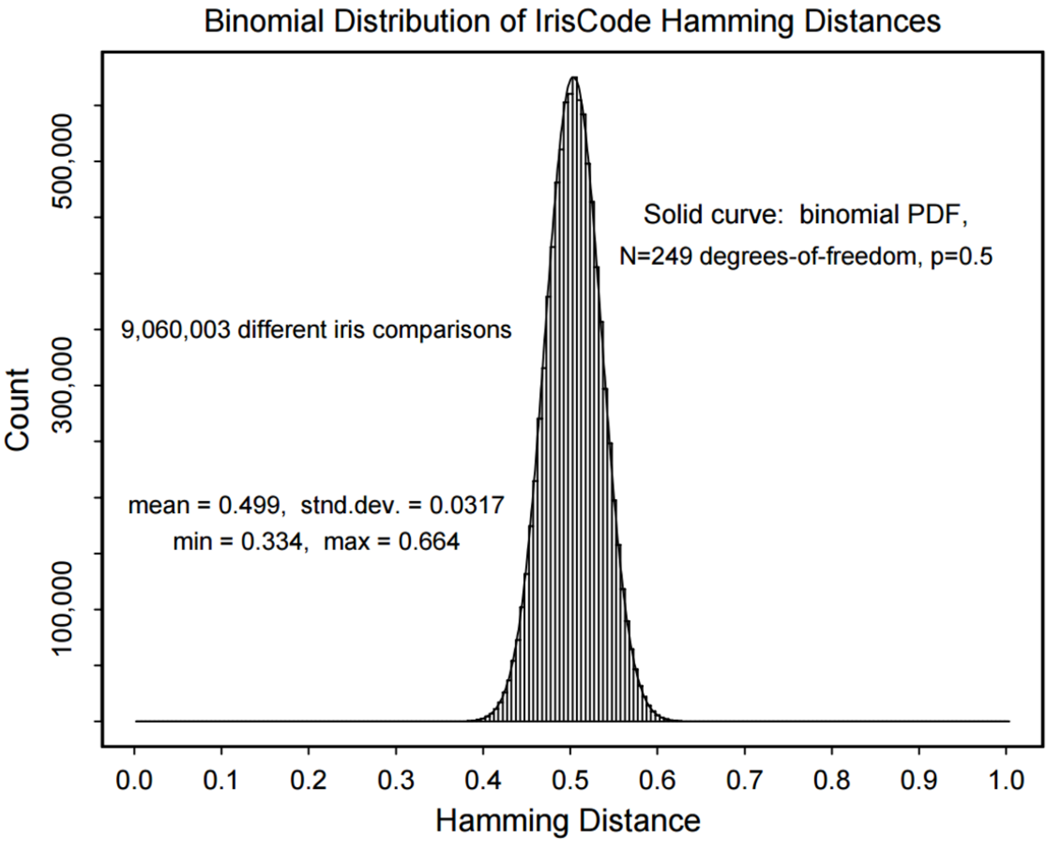


Figure 3.2: Distribution of Hamming distances of interclass comparisons for iris phase codes [1]

iris codes. We use the same methodology of fitting a binomial distribution to imposter distance data and to calculate the entropy of biohashes via the degree of freedom in the corresponding binomial distribution. A binomial distribution is fit to the obtained inter-class distances (i.e., imposter comparisons) as follows. Using the imposter comparisons between biohashes of different subjects, the observed mean of the normalized Hamming distances (μ_d) and observed standard deviation (σ_d) are calculated from data. This corresponds to a binomial distribution with $N_b = \mu_d(1 - \mu_d)/\sigma_d^2$. The theoretical binomial distribution has the functional form:

$$f(m) = \frac{N_b!}{m!(N_b - m)!}\mu_d^m(1 - \mu_d)^{(N-m)}, \tag{3.6}$$

where m/N_b ($m = 1, \ldots, N_b$) is the outcome fraction of N_b Bernoulli trials, for our case, it is the normalized Hamming distance for imposter matches. The number N_b (degree of freedom) of the binomial distribution is the predicted entropy of biohashes.

3.4 Experiments and Results

We implement the entropy estimation method described in Section 3.3.2 on a face verification set-up considering two different threat scenarios. The naive threat model assumes that an adversary has very limited information about the system and he can only perform a brute force attack using an arbitrary face information and a random secret key. In the advanced threat model, essential details of the algorithms, properties of biometric data as well as the secret keys of users are assumed to be known by the attacker. So, the attacker can create biohashes using any face image and the secret key of the user that he tries to impersonate.

3.4.1 Experimental Setup and Database

In our experiments, we use the BioSecure-ds2 [52] face database. It consists of 210 users, equally balanced by gender. 8 standard camera acquisitions per person (captured in two separate sessions) are used in our experiments. PCA coefficients extracted from detected face images are used for matching. The faces are automatically detected using Viola-Jones face detector [53] and resized to 64×64 pixels. In order to normalize a gray-scale face image, its mean intensity value is extracted from each pixel and each pixel is divided by its standard deviation.

1024-dimensional PCA coefficients are calculated for all 8 samples of 210 subjects (a total of 1680 (210×8) face images). PCA training is done using the first session images only. Applying the standard biohashing procedure, a bit-string is created through matrix-vector product between the pseudo-random matrix and 1024-dimensional PCA coefficients vector. The resulting vector is then quantized using a predefined threshold. One can obtain a bit-string of any length according to the memory and security requirements of the system. In order to demonstrate that the accuracy of the entropy analysis does not depend on biohash length, we experiment with three test lengths, namely 128, 256 and 512.

Table 3.1: Mean value, standard deviations, and degrees of freedom for different bit lengths under both scenarios

	Bit Length	**Mean (μ_d)**	**Standard Deviation (σ_d)**	**Degree of Freedom (N_b)**
Naive Model	128	0.5000	0.0443	127
	256	0.4997	0.0313	254
	512	0.5001	0.0223	504
Advanced Threat Model	128	0.3653	0.0862	31
	256	0.3685	0.0792	37
	512	0.3836	0.0761	40

3.4.2 Entropy Prediction Under Naive Threat Model

In our verification setting, all possible combinations of matching genuine pairs are used and the first sample of each subject is chosen for imposter matches (5880 $(210 \times 8 \times 7/2)$ genuine comparisons and 21945 $(210 \times 209/2)$ imposter comparisons). For imposter comparisons, the observed mean normalized Hamming distance with its standard deviation and the degree of freedom of its corresponding binomial distribution for each test length are given in Table 3.1.

The figures in the first column of Figure 3.3 illustrate the distribution of imposter distances under the naive threat model for biohashes with three test lengths. The histogram of the interclass comparison distribution (shown in blue) forms a perfect binomial distribution with parameters $\mu_d = 0.5001$, $\sigma_d = 0.0223$, and $N_b = 504$ (for 512 bits) as shown by the solid red line. The small difference between the actual bit length and the predicted entropy is due to database artifacts and it is expected that as the number of imposter comparisons gets higher, the ratio estimated entropy in bits/bit length would reach 1.

3.4.3 Entropy Prediction Under Advanced Threat Model

In the advanced threat model, the adversary is assumed to have full knowledge of the system and the secret keys of all users. The same experimental set-up of the naive model is used in order to predict the entropy of biohashes. For a biohash of a valid system user, an imposter biohash is created using the secret key of that user and a biometric template of an arbitrary user. Thus, unlike the naive model, interclass distances are calculated

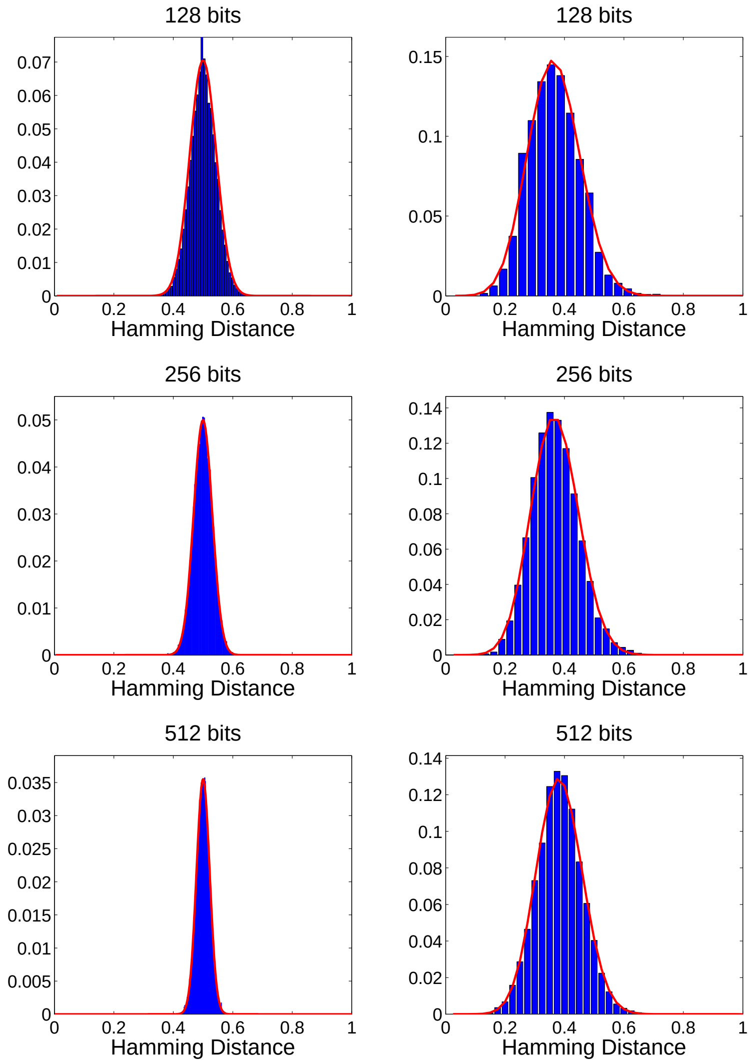


Figure 3.3: Distribution of normalized Hamming distances of interclass comparisons of biohashes with various lengths and different threat models - first column: naive threat model and second column: advanced threat model

between two biohashes that are created using the same secret key for different users. The graphs in the second column of Figure 3.3 illustrate the distribution of imposter distances for biohashes with various lengths. The observed mean of the distribution deviates from 0.5 and gets closer to the observed mean of genuine comparisons as the imposter distances get smaller. Since the distribution of genuine results is not involved in the entropy estimation, it is not discussed here. Thus, the comparison of genuine templates is not presented here for brevity.

This effect is also evident in the results given in Table 3.1. As compared to the naive model, the degree of freedom is much lower than the actual bit length and the predicted entropy decreases dramatically for all biohash lengths. For example, the entropy drops from 504 to 40 for biohashes of length 512. We argue that our results in naive and advanced scenarios are generalizable when the database is large and representative enough. For all biohash lengths, the estimated entropy in this threat model is between 31 and 40 bits which is consistent with the face entropy of 40 bits reported in [48].

3.5 Discussion

Existing theoretical evaluations of biometric protection methods cannot be used for assessing biohashing methods. In this work, we have described a systematic approach to quantify the unpredictability of random projection-based biohashing scheme by using entropy as a measure. Since feature extraction and feature normalization methods are not in our scope, we have focused only on quantitative evaluation of random projection and quantization steps. We have estimated the entropy of a biohash in terms of bits via the degree of freedom of binomial distribution under two predefined threat models [8]. Our experiments in a face verification setup have demonstrated that the entropy of a biohash is almost equal to its bit-length as expected when there is no attack against the system (the naive threat model). On the other hand, the entropy and hence the unpredictability of biohashes decrease when the attacker knows the secret key of the user that he tries to impersonate (the advanced threat model). Thus, the amount of information kept secret in a biohash becomes more likely to be predicted in such cases.

Potential future research directions on entropy of biohashes can be summarized as follows. Novel random projection methods should be studied in order to decrease the entropy loss between the naive and advanced threat models. In addition, other applicable privacy and security metrics could be investigated, such as the mutual information of hashes of different users (i.e., the entropy of one hash conditional to another hash). One other possible research direction would be to study the suitability of universal entropy estimators (e.g., Coron's or Maurer's [54]) to biohashes.

Chapter 4

Practical Security and Privacy Attacks Against Biometric Hashing Using Sparse Recovery

In this chapter, we present four different novel optimization-based methods that aim to predict the feature vector and/or the biometric image itself. Here, our assumption is that an adversary, who gains access to the biohash vector of a valid system user and the corresponding secret key, estimates a new real-valued feature vector from the binary biohash and uses it to authenticate to the system. In this study, we focus on novel feature estimation methods. The first two proposed methods are based on one-bit compressive sensing approach and related feature reconstruction algorithms. Compressive sensing is a new signal acquisition technology with the potential of reducing the number of measurements required to acquire signals that are sparse or compressible in some domain. Rather than uniformly sampling the signal, compressive sensing computes inner products with a randomized dictionary of test functions. The signal is then recovered by a convex optimization which ensures that the recovered signal is consistent with the measurements. One-bit measurements is a more restricted case in which only the sign information of the random measurements is preserved. In our framework, we solve the biohash invertibility problem by using two different reconstruction approaches, namely, linear programming [55] and binary iterative hard thresholding [56].

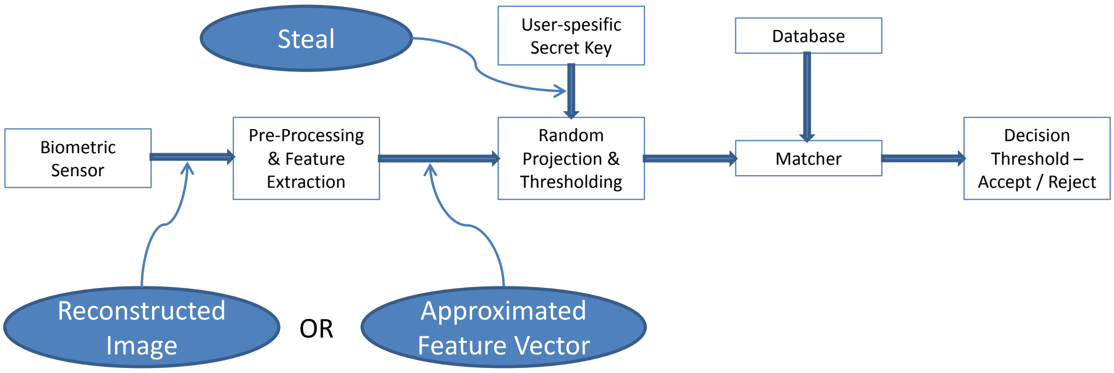

Steal
User-spesific Secret Key
Database
Biometric Sensor
Pre-Processing & Feature Extraction
Random Projection & Thresholding
Matcher
Decision Threshold – Accept / Reject
Reconstructed Image
OR
Approximated Feature Vector

Figure 4.1: Overview of a biometric hashing system

We also discuss minimum norm solutions for approximating feature vectors from biohashes and present L_2 and L_1 norm minimization for this problem. Finally, we describe the rainbow attack to compromise the security of a biometric hashing scheme. Rainbow attack is different from feature approximation methods and does not aim at predicting a new feature vector. With the help of a huge database of biometric features along with the biohash vector of a valid user and the corresponding secret key, a biometric image that creates a sufficiently close biohash to the desired one is found and used for illegitimate authentication.

We propose practical attacks and study their performances instead of using theoretical metrics. Furthermore, we analyze the privacy issues related to the invertibility of biohash templates and, as a case study, we visually inspect reconstructed face images of the subjects. Authentication performance of the reconstructed feature vectors in a conventional verification setup, in which the plain features are used for matching, is also investigated.

4.1 Proposed Feature Approximation Methods from Biohash

In this section, we introduce intrusion attacks via reconstruction of the biometric feature vector from biohashes. In this context, intrusion is defined as gaining access to a biometric recognition system by presenting falsified authentication data to the system [11] (see Figure 4.1).

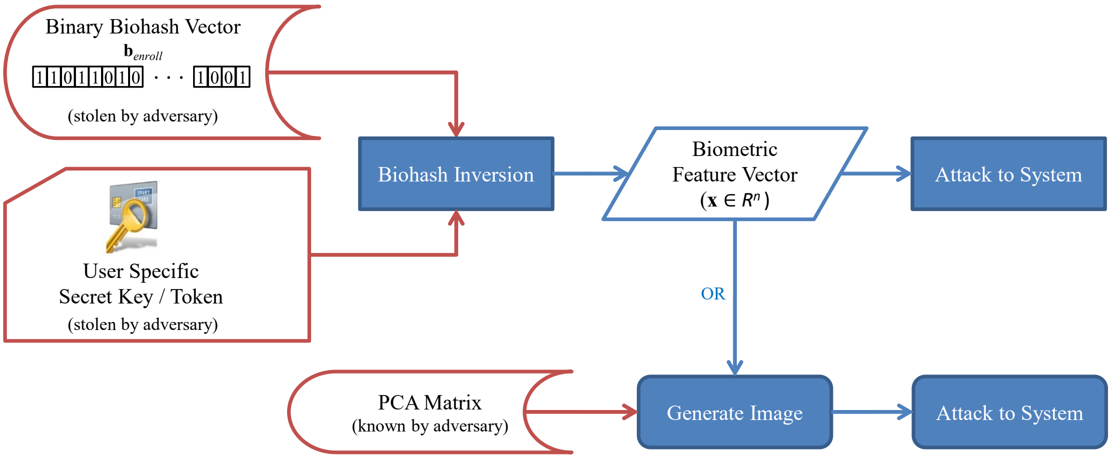


FIGURE 4.2: Illustration of the proposed attack

We use the following notation throughout our analysis of biometric hashing scheme: $\mathbf{b}$ represents the biohash vector of a valid system user and it is obtained by an adversary to perform intrusion attacks through feature approximation, $\mathbf{R}$ is the user specific random projection matrix and known to the adversary, $\mathbf{x}$ is the original biometric feature vector that $\mathbf{b}$ is created from and it is neither known nor accessible by the attacker, and $\hat{\mathbf{x}}$ is the feature vector (or pre-image) that is approximated through inversion of $\mathbf{b}$. Note that, $\hat{\mathbf{x}}$ does not necessarily correspond to a valid biometric feature vector (i.e., PCA coefficients for faces or fingerprint minutiae information). However, using $\hat{\mathbf{x}}$, one can produce a biohash vector that allows unauthorized access to the biometric system (Figure 4.2). Once $\hat{\mathbf{x}}$ is obtained, an attacker might also reconstruct the biometric modality and use it for illegitimate access to a system, i.e., in our case this is the face image (it is also assumed that the attacker knows the PCA matrix used in feature extraction). In this study, we consider that the intrusion to the system can happen in two ways before the random projection step. An attacker either provides a digital face image (reconstructed face image) to the system prior to the feature extraction step or uses the approximated feature vector as input to the random projection.

The success probability of such an attack to the biometric hashing system can be measured as $P(d(\text{sign}(\mathbf{R}\hat{\mathbf{x}}), \mathbf{b}) < \epsilon)$ [1], where $d(\cdot)$ is the Hamming distance between two biohashes (i.e., the number of disagreeing bits). This metric is also called the Intrusion Rate due to Inversion for the Same biometric system (IRIS) by Nagar et al. [11]. In

[1] This probability is estimated by using the false accept rate of the system when the threshold is ϵ

the next sections, we present various methods to obtain a feature vector $\hat{\mathbf{x}}$ that allows illegitimate access to a biometric system given **b** and the transformation parameters.

4.1.1 One-bit Compressive Sensing Approach

One-bit compressive sensing studies efficient acquisition of sparse (or more structured) signals via linear measurement systems and only 1 bit per measurement is retained. While the key application of this problem has been in the area of signal acquisition, it has also found applications in several learning related problems. Boufounos et al. [57] introduced the problem of one-bit compressive sensing where only 1 bit of the linear measurement, specifically its sign is observed. Random projection based biometric hashing can be viewed in the same context as one-bit compressive sensing. If the threshold used in quantization of the projected signal is 0 (such that the sign of the signal is kept), each bit of a biohash is the sign of the inner product of the feature vector ($\mathbf{x}$) with a measurement vector (in biometric hashing, each row of the random projection matrix ($\mathbf{R}$):

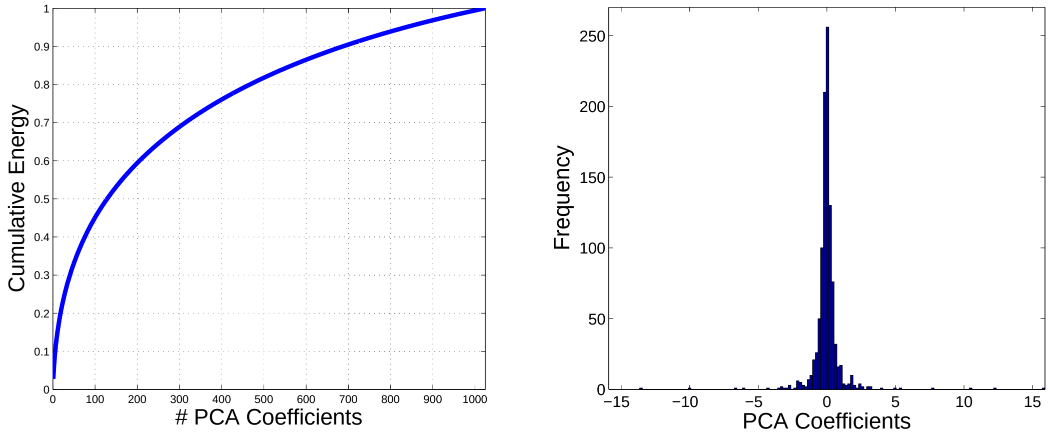
$$b_i = \text{sign}(\langle R_i, \mathbf{x} \rangle). \tag{4.1}$$

The biometric hashing procedure is compactly expressed using:

$$\mathbf{b} = \text{sign}(\mathbf{R}\mathbf{x}), \tag{4.2}$$

where **b** is the biohash vector, **R** is the matrix representing the random projection matrix (the measurement system), and the 1-bit quantization function sign(.) is applied element-wise to the vector $\mathbf{Rx}$.

For consistent reconstruction from 1-bit measurements, the measurements are treated as sign constraints that are enforced in the reconstruction to recover the signal. In the reconstruction, L_1 norm of the feature vector is minimized to obtain a sparse solution. When the PCA coefficients of face images are analyzed, it is noted that most of the coefficients are small in magnitude and only about 25% of them is enough to obtain $\sim$ 70% of the total energy as seen in Figure 4.3. Therefore, it is reasonable to assume that PCA vectors are sparse. Also, as stated by Candes and Wakin [58], "compressive



(a) (b)

FIGURE 4.3: (a) Cumulative energy contained in PCA coefficients. (b) Distribution of 1024 dimensional PCA coefficients of a sample from the database.

sampling exploits the fact that many natural signals are sparse in the sense that they have concise representations when expressed in the proper basis". Even if the original signal is not sparse, a basis can be found in which most coefficients are small, and the relatively few large coefficients capture most of the information and this allows for the use of sparse recovery in the problem of biohash inversion.

In addition, to enforce reconstruction at a non-trivial solution, one needs to artificially resolve the amplitude ambiguity. Thus, an energy constraint is imposed that the reconstructed signal lies on the unit L_2-sphere:

$$\|\mathbf{x}\|_2 = \left(\sum_i x_i^2\right)^{1/2} = 1. \tag{4.3}$$

The signal on the unit sphere that is consistent with the measurements is found by solving:

$$\begin{aligned} &\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \\ &\text{s.t. } \operatorname{sign}(\mathbf{R}\hat{\mathbf{x}}) \equiv \mathbf{b} \\ &\text{and } \|\hat{\mathbf{x}}\|_2 = 1. \end{aligned} \tag{4.4}$$

As the compressive sensing measurements are quantized to one bit, it is clear that the scale (absolute amplitude) of the signal is lost and it is not immediately evident that the remaining information is enough for signal reconstruction. Nonetheless, there is strong empirical evidence stating that signal reconstruction is possible [57]. One-bit compressive sensing by linear programming [55] and binary iterative hard thresholding [56] are two theoretical reconstruction methods that we implement separately for obtaining inverse images of biohashes and finding biometric feature vectors that provide biohash vectors which are acceptable by the verification system (i.e., with a distance to the original biohash vector that is less than a threshold).

4.1.1.1 One-bit Compressive Sensing by Linear Programming

The study in [55] has showed that $\mathbf{x}$ can be accurately estimated from extremely quantized measurement vector in (4.2). Note that, $\mathbf{b}$ contains no information about the magnitude of $\mathbf{x}$ and only the normalized vector $\mathbf{x}/\|\mathbf{x}\|_2$ can be recovered. It has been shown that the signal can be accurately recovered by solving the following convex minimization program:

$$\begin{aligned} &\min \|\hat{\mathbf{x}}\|_1 \\ &\text{s.t. } \mathbf{B}\mathbf{R}\hat{\mathbf{x}} \geq \mathbf{0} \\ &\text{and } \|\mathbf{R}\hat{\mathbf{x}}\|_1 = m, \end{aligned} \tag{4.5}$$

where $\mathbf{B} = \mathrm{diag}(\mathbf{b})$.

The first constraint, $\mathbf{B}\mathbf{R}\hat{\mathbf{x}} \geq \mathbf{0}$, keeps the solution consistent with the original biohash vector and it is defined by the relation $\langle R_i, \hat{\mathbf{x}}\rangle \cdot b_i \geq 0$ for $i = 1, 2, \ldots, m$ where R_i is the i^{th} row of the random projection matrix $\mathbf{R}$. The second constraint in the original problem definition (4.4) contains L_2-norm which is a quadratic term and can be replaced with the linear L_1-norm, so that the optimization becomes a linear program. The second constraint, $\|\mathbf{R}\hat{\mathbf{x}}\|_1 = m$, serves to prevent the program from returning zero solution and it is linear as it can be represented as one linear equation $\sum_{i=1}^{m} b_i\langle R_i, \hat{\mathbf{x}}\rangle = m$, where m is the length of the biohash vector. Therefore, (4.5) is a convex minimization problem and can easily be represented as a linear program (see Algorithm 4.1).

Algorithm 4.1 Approximate biometric feature vector $\hat{\mathbf{x}}$ using Linear Programming

```
Input: $\mathbf{b}$, $\mathbf{R}$
Output: $\hat{\mathbf{x}}$
  calculate $\mathbf{A}$ such that $\mathbf{A}\hat{\mathbf{x}} \geq \mathbf{0}$ using $\mathbf{b}$ and $\mathbf{R}$
  calculate $\mathbf{A}_{eq}$ such that $\mathbf{A}_{eq}\hat{\mathbf{x}} = m$ using $\mathbf{R}$
  set $f$ to calculate $L_1$ norm of $\hat{\mathbf{x}}$
  use simplex method to solve for $\hat{\mathbf{x}}$
```

4.1.1.2 Binary Iterative Hard Thresholding

Binary iterative hard thresholding (BIHT) [56] is a modification of iterative hard thresholding (IHT) which is a real-valued algorithm designed for compressive sensing [59]. Proposed for the recovery of K-sparse signals, IHT algorithm consists of two steps. The first step is a gradient descent to reduce the least squares objective $\|\mathbf{y} - \mathbf{R}\mathbf{x}\|_2^2/2$. At each iteration, IHT proceeds by setting $\mathbf{a}^{l+1} = \mathbf{x}^l + \mathbf{R}^T(\mathbf{y} - \mathbf{R}\mathbf{x})$. The second step imposes a sparse signal model by selecting the K elements of $\mathbf{a}^{l+1}$ that are largest in magnitude.

BIHT algorithm modifies the first step of IHT and minimizes a consistency-enforcing objective. Given an initial estimate $\mathbf{x}^0 = \mathbf{0}$ and 1-bit measurements $\mathbf{b}$, at each iteration l, BIHT computes:

$$\begin{aligned} \mathbf{a}^{l+1} &= \mathbf{x}^l + \frac{\tau}{2}\mathbf{R}^T(\mathbf{b} - \operatorname{sign}(\mathbf{R}\mathbf{x}^l)) \\ \mathbf{x}^{l+1} &= \eta_K(\mathbf{a}^{l+1}), \end{aligned} \tag{4.6}$$

where τ is a scalar that controls the gradient descent step size, and the function η_K computes the best K-term approximation of $\mathbf{a}^{l+1}$ (see Algorithm 4.2). In our experiments, we choose K as 25% of the feature vector length, i.e., $K = 50$ for 200 dimensional feature vectors and $K = 256$ for 1024 dimensional feature vectors.

Algorithm 4.2 Approximate biometric feature vector $\hat{\mathbf{x}}$ using BIHT

```
Input: $\mathbf{b}$, $\mathbf{R}$, $K$
Output: $\hat{\mathbf{x}}$
  initialize $\mathbf{x}^0$ all zeros
  while $|\mathbf{b} - \operatorname{sign}(\mathbf{R}\mathbf{x}^l)|_1 > 0$ do
    $\mathbf{a}^{l+1} = \mathbf{x}^l + \frac{\tau}{2}\mathbf{R}^T(\mathbf{b} - \operatorname{sign}(\mathbf{R}\mathbf{x}^l))$
    sort elements of $\mathbf{a}^{l+1}$ and set the all but the largest $K$ components to 0,
  end while
  set $\hat{\mathbf{x}} \leftarrow \mathbf{a}^{l+1}$
```

4.1.2 Minimum L_1 and L_2 Norm Solutions

In this section, we present and discuss minimum-norm based feature reconstruction methods for biohashes in addition to the solutions we propose in one-bit compressive sensing framework.

Biohash vector is obtained through quantization from an intermediate vector $\mathbf{z}$ which is the output of a random projection, i.e., $\mathbf{z} = \mathbf{Rx}$. If one can invert the quantization step and find an intermediate vector $\hat{\mathbf{z}}$ that produces the biohash vector after quantization, a minimum norm solution can be used to estimate the biometric feature vector ($\hat{\mathbf{x}}$) as:

$$\min \|\hat{\mathbf{x}}\|_n \text{ s.t. } \hat{\mathbf{z}} = \mathbf{R}\hat{\mathbf{x}}. \tag{4.7}$$

In this work, we study minimum norm solutions for $n = 1$ and $n = 2$, namely L_1 and L_2 norms.

4.1.2.1 Inversion of the Quantization Step

Solutions in one-bit compressive sensing framework implicitly handle the quantization of the randomly projected feature $\mathbf{z}$ within the optimization process. However, L_1 and L_2 norm-based reconstruction requires an explicit inversion of the thresholding step of the biometric hashing scheme.

In order to invert the quantization process, an adversary who possesses the biohash ($\mathbf{b}$) of a valid system user and corresponding random projection matrix ($\mathbf{R}$), uses an arbitrary biometric feature vector $\mathbf{x}_f$ to simulate the biometric hashing procedure through random projection and obtain an intermediate vector $\mathbf{z}_f = \mathbf{R}\mathbf{x}_f$. Next, the sample mean and standard deviation of $\mathbf{z}_f$ are calculated, μ and σ respectively. Mapping the elements of the compromised biohash vector $\mathbf{b}$ from {0,1} to {-1,1} is performed as:

$$\hat{b}(i) = \begin{cases} 1, & b(i) = 1, \\ -1, & b(i) = 0, \end{cases} \tag{4.8}$$

where $\hat{\mathbf{b}}$ is the mapped biohash vector. Finally, using the values calculated from the arbitrary biometric features, the intermediate vector $\hat{\mathbf{z}}$ is estimated as:

$$\hat{z}(i) = \mu + \hat{b}(i)\sigma. \tag{4.9}$$

To be consistent with the solutions described in one-bit compressive sensing approach, we assume that the signs of elements of the intermediate vector $\mathbf{z}$ is used to obtain the biohash (i.e., the threshold at the quantization step is 0). However, various quantization methods and thresholding mechanisms are proposed in the literature for biometric hashing, one of them being the mean value of the intermediate vector and another one being its median value. If the system uses the mean value of the intermediate vector as the quantization threshold, the mean value of the $\mathbf{z}_f$ can be calculated. In our experiments, the threshold equals to 0, thus the mean value is not used, and the intermediate vector is computed as $\hat{z}(i) = \hat{b}(i)\sigma$.

4.1.2.2 Minimum L_2 Norm Solution

Once an adversary creates an intermediate vector $\hat{\mathbf{z}}$, the following L_2 norm minimization provides an estimate feature vector $\hat{\mathbf{x}}$ that is consistent with the observation $\mathbf{b} = \text{sign}(\mathbf{R}\hat{\mathbf{x}})$.

$$\min \|\hat{\mathbf{x}}\|_2 \text{ s.t. } \hat{\mathbf{z}} = \mathbf{R}\hat{\mathbf{x}}. \tag{4.10}$$

The closed form solution that gives the minimum L_2 norm for the estimated feature vector is given by the MoorePenrose pseudo-inverse. For linear systems $\mathbf{Ax} = \mathbf{b}$ with non-unique solutions (i.e., under-determined systems), the pseudo inverse is used to reconstruct the solution of minimum Euclidean norm $\|\mathbf{x}\|_2$ among all solutions. So the solution to the above minimization problem to estimate the feature vector from biohash $\mathbf{b}$ is calculated as $\hat{\mathbf{x}} = \mathbf{R}^{\dagger}\hat{\mathbf{z}}$.

4.1.2.3 Minimum L_1 Norm Solution

Similar to the minimum L_2 norm solution, minimum L_1 norm solution aims at solving the following minimization problem.

$$\min \|\hat{\mathbf{x}}\|_1 \text{ s.t. } \hat{\mathbf{z}} = \mathbf{R}\hat{\mathbf{x}}. \tag{4.11}$$

In one-bit compressive sensing approach by linear programming, L_1 norm of the estimated feature vector is minimized according to the constraints that include the quantization step. However, minimum L_1 norm solution handles the quantization step separately and the minimization is carried out over the intermediate real-valued vector $\hat{\mathbf{z}}$. The minimization problem still has linear constraints and minimization of L_1 norm can easily be expressed as a linear program and solved accordingly.

For both L_1 and L_2 norm minimizations, if the PCA dimension is less than the biohash length (i.e., if the random projection step does not reduce the dimension), the linear system is over-determined and an exact solution might not possibly exist (i.e., solutions could be inconsistent with the observations). Instead, it is possible to minimize the residual between the observation and the solution (i.e., $\|\hat{\mathbf{z}} - \mathbf{R}\hat{\mathbf{x}}\|_n$) and to obtain a feature vector that provides biohashes that is close to the original one.

4.1.3 Reconstructing the Face Image

As long as the feature extraction step uses an orthogonal transformation matrix, it is possible to invert the feature extraction process simply by using the pseudo inverse of the transformation matrix and a face image can be reconstructed easily. The Principal Component Analysis uses an orthogonal transformation, which means that the columns of the PCA matrix are perpendicular to each other and hence one can reconstruct the face image $\hat{\mathbf{y}}$ from $\hat{\mathbf{x}}$ by using the property of an orthogonal matrix $\mathbf{A}^\dagger = \mathbf{A}^T$:

$$\hat{\mathbf{y}} = \mathbf{A}^T\hat{\mathbf{x}} + \mu_{\mathbf{y}}, \tag{4.12}$$

where $\mathbf{A} \in \Re^{k\times(mn)}$ is the PCA matrix, $\mathbf{A}^\dagger$ is the pseudo-inverse of $\mathbf{A}$, and $\mu_{\mathbf{y}}$ is the mean face vector.

4.1.4 Other Thresholding Methods - Apart from the "sign" Operator

In cases where the thresholding after the random projection step is not the sign operator, some alternatives can also be formulated within our proposed framework. Assuming that an adversary has the full knowledge of the system, i.e., the specific thresholding method, he can also invert the biohashes.

4.1.4.1 Fixed or User Specific Threshold

Apart from using the sign operator, one can use a pre-defined fixed threshold or user specific threshold, i.e., $\mathbf{b} = \text{sign}(\mathbf{Rx} - \mathbf{T})$ where $\mathbf{T}$ denotes the threshold. Entries of $\mathbf{T}$ can be the same number or different numbers at each dimension. $\mathbf{T}$ can also be specific to each user (it is show as $\mathbf{T}_i$ where i denotes the subject number). By augmenting the threshold vector to the random projection matrix, $\hat{\mathbf{R}} = \begin{bmatrix}\mathbf{R} & -\mathbf{T}_i\end{bmatrix}$, we can reformulate the biohashing operation as $\mathbf{b} = \text{sign}\left(\hat{\mathbf{R}}\begin{bmatrix}\mathbf{x} & 1\end{bmatrix}\right)$ and perform the same operations for inverting biohashes.

4.1.4.2 Mean Value is the Threshold

An alternative way of thresholding the intermediate vector is to use the mean value of the intermediate biohash vector $\mathbf{z} = \mathbf{Rx}$ as the threshold and to calculate the biohash vector as

$$\mathbf{b} = \text{sign}(\mathbf{Rx} - \text{mean}(\mathbf{Rx})). \tag{4.13}$$

Thresholding step can be integrated into the random projection step by using the modified random projection matrix $\hat{\mathbf{R}}$:

$$\hat{\mathbf{R}} = \left[\mathbf{R} - \frac{\mathbf{1}\cdot\mathbf{R}}{N}\right], \tag{4.14}$$

where N is the biohash length, $\underline{\mathbf{1}}$ is a matrix of ones, and the biohash vector becomes $\mathbf{b} = \text{sign}(\hat{\mathbf{R}}\mathbf{x})$. An adversary can use the modified matrix $\hat{\mathbf{R}}$ and all inversion methods that we discuss are still valid in this setup.

4.2 Rainbow Attack

In the previous section, we propose four different optimization methods for recovering features from an original biohash vector that is stolen by an attacker. Having the corresponding secret key and using the knowledge of system parameters, one can estimate a real valued feature vector $\hat{\mathbf{x}}$ with the consistency criterion such that $\mathbf{b} = \text{sign}(\mathbf{R}\hat{\mathbf{x}})$ in order to gain illegitimate access to the biometric system. Rainbow attack is different from these methods in the sense that it does not aim at inverting a biohash vector to obtain a valid pre-image. Instead, using the knowledge of the system and the secret key of the user, with the help of a large database of biometric features, an adversary may find a face image which, when combined with the secret key of the user, result in a biohash vector that is sufficiently close to the original biohash $\mathbf{b}$.

In the cryptography literature, a rainbow table is a precomputed table for reversing cryptographic hash functions, usually for cracking password hashes. Any computer system that requires password authentication must contain a database of passwords, either hashed or in plaintext, and utilize different methods to store passwords. Because the tables are vulnerable to thefts, storing passwords as plain texts is dangerous. Most databases therefore store a cryptographic hash of a user's password in the database. When a user enters his password for authentication, it is hashed and compared to the stored password entry of that user (which is also hashed before being stored in the database). If the two hashes match, the access is granted. A Rainbow Table is a large dictionary with pre-calculated hashes and the passwords from which they were calculated. When an attacker steals a long list of password hashes from the system, he can quickly check if any of them are in the Rainbow Table. If that is the case, the Rainbow Table will also contain the original string that they were hashed from.

A biometric authentication system that protects biometric templates using biometric hashing methods operates in a similar way; the biohash of a user is stored and compared to the query biohash during verification. If an adversary having a large database of biometric features of various users, steals the biohash of a system user and knows his secret key, the adversary can compute biohashes of each biometric feature in the database using the random projection matrix of the user and create a table of biohashes and their corresponding feature vectors. If any of the biohashes in the table is sufficiently close to the stolen biohash (i.e., their Hamming distance is less than a threshold), the corresponding feature vector can be used for illegitimate access to the biometric system.

Different from previously described attacks which try to approximate a feature vector that gives a close biohash vector to the stolen one, the rainbow attack is a practical attack that aims to compromise the security of a biometric hashing scheme. Furthermore, assuming that one authentication factor (the secret key of a user) is known, the rainbow attack also provides privacy threat since look alike faces can be found.

4.3 Experiments and Results

In this section, the performance of our proposed attack methods are analyzed and discussed. The database that is used and the experimental set-up are described, and attack models and their corresponding error rates are given.

4.3.1 The Database and Experimental Set-up

In order to provide the performance analysis of the security of biohashes based on the feature approximation methods, we implement our proposed methods on a face verification setup. We obtain face verification results on the BioSecure-ds2 face database [52]. The same database set-up with Section 3.4 is followed. M-dimensional PCA coefficients are calculated for all 8 samples of 210 subjects. In our results, we present results using bit-strings of length 128, 256, 512 and 1024.

In a verification setting, we use all possible combinations for matching genuine pairs and the first sample of each subject is chosen for imposter matches (5880 $(210 \times 8 \times 7/2)$ genuine comparisons and 21945 $(210 \times 209/2)$ imposter comparisons) in order to evaluate the performance of the biometric hashing scheme. For validating the consistency of approximated features using the proposed methods, we compare the biohashes created from these features with the original biohashes leading to one imposter score for each sample in the database (1680 imposter matches). Equal error rates (EER) in each case are reported.

4.3.2 The Performance of the Biometric Hashing Scheme

First, we apply the general biometric hashing scheme described in Chapter 3 on the BioSecure-ds2 face database. For comparison, we also include face verification results of PCA vectors by using Euclidean distance as the matching method. The equal error rates for this method before applying biometric hashing to PCA vectors are 11.893% and 12.482% for vectors of length 200 and 1024, respectively. Equal error rates for biohash vectors of various lengths are given in Table 4.1.

Table 4.1: Equal Error Rates (%) for biohash vectors of different lengths

	PCA 200		**PCA 1024**	
Bit Length	**Biohash**	**Biohash (Stolen Key)**	**Biohash**	**Biohash (Stolen Key)**
128	6.295	12.571	6.593	13.565
256	4.570	11.457	4.813	12.216
512	4.137	11.595	4.328	11.634
1024	2.875	11.118	2.934	11.553

For all bit lengths, the performance of the biometric hashing scheme is better than the baseline PCA approach and lower EERs are obtained with the protected templates. In cases where an adversary steals the secret key of a user but does not possess the claimed person's biometric information, the adversary sends his own biometric (or an arbitrary biometric) and the secret key of the genuine user in order to be authenticated. This is a serious threat to the system as the pseudo-random vectors generated using the secret key have a considerable influence on the generated bit string, therefore on the matching score. However, even if the attacker knows the secret key, the verification accuracy of the biometric hashing system is still in the same range with the performance of the unprotected PCA vectors.

One obvious addition to the biometric hashing scheme is the direct comparison of secret keys (i.e., the one stored during enrollment and the one presented during authentication) prior to biohash comparison. This way 0% (zero) EER is achieved if the attacker does not have the secret key of a valid user. The error rates presented in Table 4.1 are the results of biohash comparison and if key checking mechanism is applied as illustrated in Figure 3.1, the EERs for the first scenario would be 0%. So that, here we study the added security coming from the biometrics with the use of biohashes in cases where an attacker obtains the secret key.

4.3.3 The Performance of the Feature Approximation from Biohash Methods

Since the database that we use has 1680 samples from 210 subjects, using their PCA coefficients and secret keys of each subject, we create 1680 biohashes, each corresponding to a different sample. It is assumed that an adversary obtains the biohash and the secret key of a user and with this knowledge he aims to find a feature vector by inverting the biohash. With this new feature vector, a new biohash can be calculated and used for authentication purposes. For each biohash in the database, we obtain a new feature vector and create its corresponding biohash. We use the new biohash to perform an imposter attack to the original one and we do not attack to other genuine samples. We use all possible combinations to match genuine pairs (5880 ($210 \times 8 \times 7/2$)) and the number of imposter comparisons is 1680 (one for each biohash). The performance of each method is reported in terms of the equal error rate (EER) and higher EER shows the success of the attacker (i.e., 100% EER means that the inversion of all biohashes in the database is successful and the approximated features provide biohash that matches with the original one).

In order to evaluate the security that biometric hashing provides, we follow three consecutive scenarios:

Advanced Attack Model (AAM): The attacker, who knows the system details and possesses the biohash of a user and his secret key, calculates an estimate feature vector. Using this feature vector and the secret key of the subject, a new biohash is created and compared with the original one.

Security After Key Change (SAKC): Upon the detection of a security breach, the secret key of the user is changed by the system administrator. Using the previous biometric data, a new biohash is created from the new secret key and stored as the new gallery template in the system. The adversary does not have access to neither the new secret key nor the new biohash. The adversary makes an authentication attempt using the feature vector found in the advanced attack model and the previous (or an arbitrary) secret key. It should be noted that, these errors are available only when the system does not perform key checking prior to biohash comparison. As the attacker does not know the secret key of the user, the EER in a key-checking system is 0%.

However, for the sake of completeness, a no key-checking system is also considered and EERs in this case are also reported. EERs presented in Table 4.2 correspond to the attack in which the adversary has the true (original) biometric features but does not possess the associated secret key. These numbers provide a lower bound on the long-term security error, where the secret key of the user is changed and is not known to the attacker.

Table 4.2: Equal Error Rates (%) when the adversary has the true biometric features but does not possess the associated secret key

PCA dimension	**Biohash Length**			
	128	**256**	**512**	**1024**
200	6.199	4.290	4.243	2.917
1024	6.497	4.902	4.375	3.044

Attack in the Long-term (ALT): The adversary obtains the new secret key of the user but not the new biohash. Using the feature vector found in the advanced attack model and the new secret key, the adversary makes an authentication attempt. This is different from the advanced attack model in the sense that the biohash vector of the user is not known to the adversary and the authentication attempt is performed using the approximated feature vector which is obtained from the previous biohash of the user.

4.3.3.1 Results for One-bit Compressive Sensing Approaches

We use two different feature approximation methods, namely linear programming (LP) and binary iterative hard thresholding (BIHT), in the one-bit compressive sensing framework. The success rates of both methods are presented in Table 4.3 and Table 4.4. For the advanced attack model, the number of exact reconstructions, i.e., the number of estimated features that provide the exact same biohashes (such that the Hamming distance between the original biohash and the forged biohash is 0) is 1680 for all bit lengths. For every sample in the database, regardless of the PCA dimension, both methods are able to find a feature vector that provides the exact same biohash and that is also reflected by 100% EERs.

Table 4.3: Equal Error Rates (%) for one-bit compressive sensing approaches - linear programming (LP) method

PCA	Bit Length	AAM	SAKC	ALT
200	128	100.00	7.262	48.333
	256	100.00	5.225	65.570
	512	100.00	4.018	78.958
	1024	100.00	3.308	89.987
1024	128	100.00	7.530	40.187
	256	100.00	5.128	53.342
	512	100.00	4.286	68.907
	1024	100.00	3.444	80.863

Table 4.4: Equal Error Rates (%) for one-bit compressive sensing approaches - BIHT method

PCA	Bit Length	AAM	SAKC	ALT
200	128	100.00	7.381	33.767
	256	100.00	4.851	49.388
	512	100.00	3.958	74.809
	1024	100.00	3.367	90.536
1024	128	100.00	6.667	16.314
	256	100.00	5.306	19.887
	512	100.00	4.252	28.759
	1024	100.00	3.474	47.653

In the security after key change scenario, when the secret key of the user is changed but not known to the adversary, EERs are in the same line with the cases where the adversary has access only to one of the factors, either true biometric or true secret key (see Tables 4.1 and 4.2). In the attack in the long term (ALT) scenario, it is possible for the attacker to have unauthorized access to the system most of the time, especially if the PCA length is shorter and the biohash length is longer (see the ALT column in Tables 4.3 and 4.4).

Values of the intermediate vector $\mathbf{z} = \mathbf{R}\mathbf{x}$ which are very close to the threshold, e.g., values $\mathbf{z}$ that are close to zero for the sign operator, lead to numerical inconsistencies about the inequality criteria of the linear program (i.e., $\mathbf{B}\mathbf{R}\mathbf{x} \geq 0$) and can be solved by replacing the inequality constraint with $\mathbf{B}\mathbf{R}\mathbf{x} \geq \epsilon$, where ϵ is the minimum positive number available in MATLAB (machine epsilon).

4.3.3.2 Results for Minimum Norm Solutions

The same set of experiments on the invertibility of biohashes is conducted using the proposed minimum norm solutions (see Table 4.5 and Table 4.6). For biohashes created from PCA vector of length 1024, both methods are able to find a pre-hash vector that can be used to create the same biohash for each sample in the database. As in the one-bit compressive sensing approach, the number of exact reconstructions is also 1680 in this case. However, when less number of PCA coefficients are used in the system (i.e., the PCA feature vectors are 200 dimensional), there is a slight decrease in the equal error rates. Biohashes created from the estimated feature vectors are not exactly same with the original ones (i.e., the Hamming distance between them is greater than zero) which is reflected by the slight deviation from 100% EER.

Table 4.5: Equal Error Rates (%) for minimum norm solutions - L_2 norm

PCA	Bit Length	AAM	SAKC	ALT
200	128	100.00	7.113	31.233
	256	99.843	5.196	34.753
	512	99.239	4.018	72.513
	1024	98.444	3.219	86.599
1024	128	100.00	7.117	17.623
	256	100.00	5.544	21.003
	512	100.00	4.256	28.703
	1024	100.00	3.474	36.947

Table 4.6: Equal Error Rates (%) for minimum norm solutions - L_1 norm

PCA	Bit Length	AAM	SAKC	ALT
200	128	100.00	6.815	30.965
	256	97.113	5.106	28.563
	512	92.491	3.839	61.173
	1024	92.751	3.431	77.564
1024	128	100.00	6.577	17.534
	256	100.00	5.723	20.765
	512	100.00	4.196	28.346
	1024	100.00	3.474	36.947

In the SAKC scenario, the performances of minimum norm solutions are similar to the one-bit compressive sensing solutions. If the new key of the user is stolen (the ALT scenario), one-bit compressive sensing approaches provide significantly higher error rates which shows the success of the attack method.

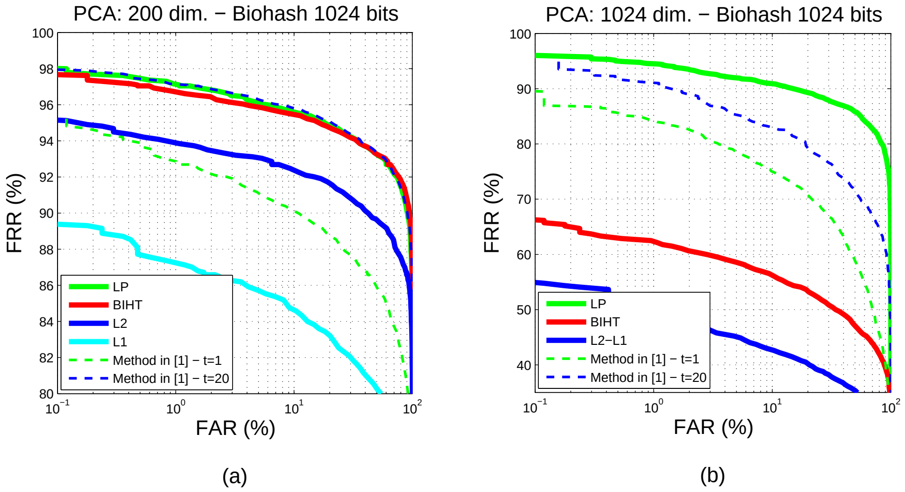


FIGURE 4.4: DET curves for the proposed methods under the scenario Attack in the Long-Term. (a) Reconstruction of 200 dimensional PCA feature vectors from biohash of length 1024-bits. (b) Reconstruction of 1024 dimensional PCA feature vectors from biohash of length 1024-bits.

Figure 4.4 illustrates the detection error tradeoff curves for the attacks using the proposed methods under the ALT scenario (together with the results of the the study in [11]). Table 4.7 shows the corresponding FAR1000 values (False Reject Rates when the FAR $= 10^{-3}$). The attack performance of the reconstructed feature vectors from biohashes of 1024-bits can be compared among different methods. For brevity, we do not include all results for different biohash lengths.

TABLE 4.7: FAR1000 values for the proposed methods under the scenario Attack in the Long-Term.

Method	**200 → 1024**	**1024 → 1024**
LP	97.9932	95.9864
BIHT	97.6190	66.1565
L_2	94.1190	54.7789
L_1	89.3027	54.7789

A special case of solving the norm-minimization problem is when the PCA feature vector dimension is equal to the length of biohash in bits. In approximating the 1024 dimensional PCA vector from biohash of length 1024 bits, there is a single unique solution. However, the condition number of the random projection matrix is so high and this leads to inaccurate solutions. To improve the solution by decreasing the condition number, we add the 20% of the maximum singular value of the matrix $\mathbf{R}$ to all singular values. This way, the condition number of $\mathbf{R}$ decreases by $\sim 10^2$.

4.3.3.3 Computation Times for the Proposed Feature Approximation Methods

The proposed feature approximation methods are implemented in MATLAB and the experimental results are run on a 2.5 GHz with 64 GB of RAM PC using 64-bit Windows Server 2008 operating system. From a given biohash of length 1024-bits and the corresponding secret key, we estimate the PCA feature vectors with four proposed methods, for PCA dimensions of 200 and 1024, respectively (Table 4.8). It is intuitive that for all methods it is faster to estimate a 200-dimensional feature vector. Among the four proposed methods, L_2-norm minimization is the first to estimate a 200-dimensional feature vector from a biohash of length 1024-bits. On the other hand, when the feature vector to be estimated is 1024-dimensional, the BIHT method performs faster than other methods.

Table 4.8: Computation time required to estimate a feature vector from a given biohash (in seconds)

Method	**1024 → 200**	**1024 → 1024**
LP	12.681736	144.288818
BIHT	0.192342	0.294719
L_2	0.108523	1.681796
L_1	11.451703	26.453929
Method in [11] for t = 1	28.244039	185.517469
Method in [11] for t = 20	572.584385	4700.410120

4.3.4 Results for the Rainbow Attack

The rainbow attack is different from feature approximation methods and its success mainly depends on the availability of a huge biometric database. In this study, we simulate the rainbow attack where an adversary has the secret key and the biohash of the user. We use the BioSecure-ds2 database and take the attacked user out of the set. We calculate the biohashes of the remaining face images with the secret key of the user and search for the one that is closest to the user's biohash. In this manner, we describe three different scenarios:

Table 4.9: Equal Error Rates (%) for the rainbow attack

PCA	Bit Length	CM	SAKC	ALT
200	128	53.597	6.964	38.571
	256	49.787	4.762	40.179
	512	47.177	4.043	41.820
	1024	46.054	3.342	43.469
1024	128	56.467	7.440	38.746
	256	51.786	5.795	41.417
	512	48.206	4.524	42.543
	1024	46.794	3.296	43.439

Collusion Model (CM): Keys are known to the attacker and using an available database, he finds the faces that provide the closest biohash given the secret key of the valid user.

Security After Key Change (SAKC): Secret keys of users are changed by the system administrator. The attacker does not know the new key but uses the face found in the CM scenario.

Attack in the Long-term (ALT): The attacker obtains the new key. He uses the face found in the CM scenario and the new key to create biohashes.

The equal error rates for the rainbow attack on biohashes for these three scenarios are given in Table 4.9. Our visual inspection shows that, faces which create close biohashes when combined with the same secret key are visually alike. This should also be regarded as a threat to the privacy of the user, as well as a threat to the security of the system (Figure 4.5).

4.3.5 Privacy Assessment of the Proposed Methods

A critical implication of the reversibility of biohashes is the relation between the reconstructed feature vectors and the original biometric information (face) of the users. For assessing to what extent the privacy of the user is at stake if his/her biohash is inverted via our proposed methods, we compare face images reconstructed using the original PCA vectors and the estimated features. Assuming that the attacker knows the details of feature extraction (PCA matrix and mean vector), we reconstruct face images with

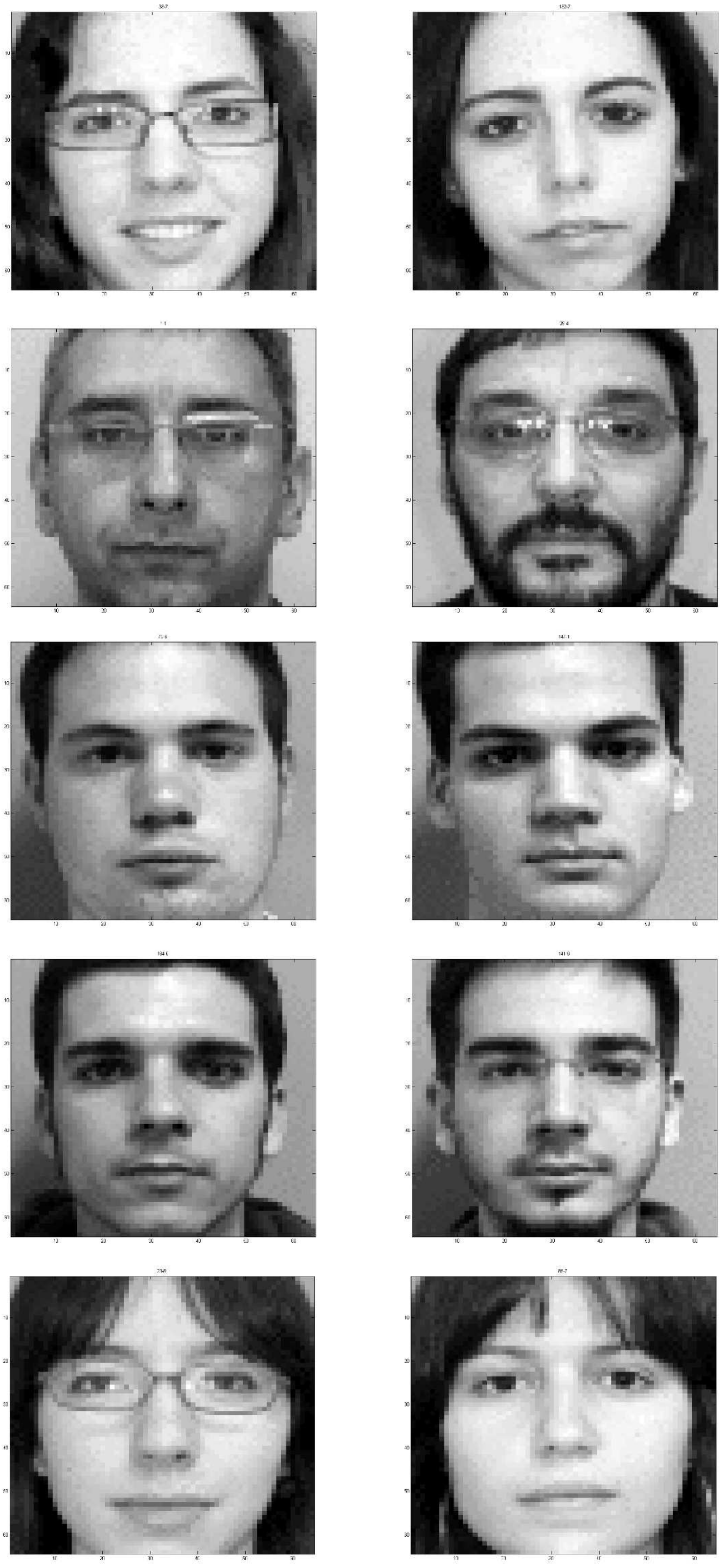
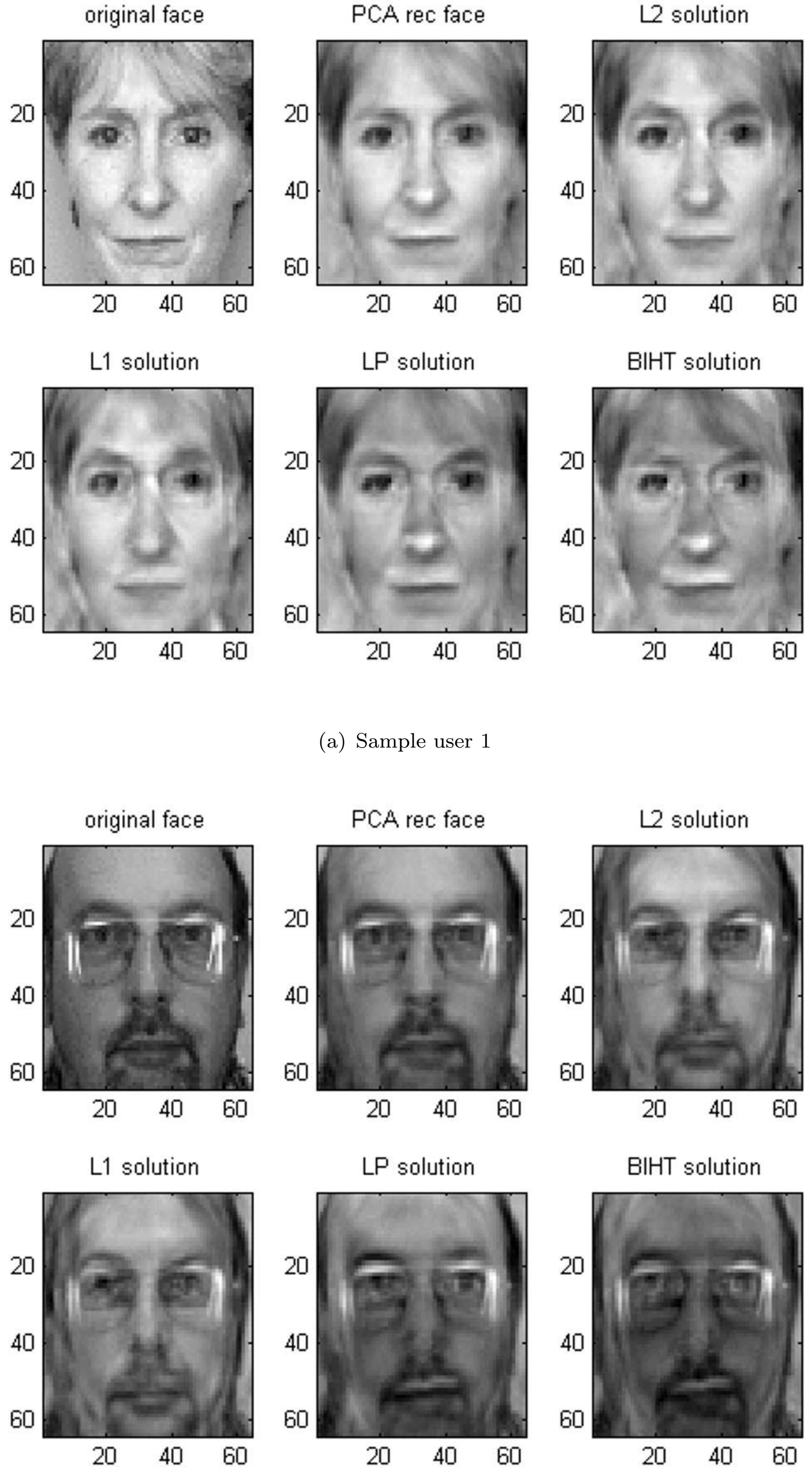


Figure 4.5: Rainbow attack - faces that provide close biohashes.

the approximated feature vectors using (4.12). In the following figures (Figures 4.6 and 4.7), we present the original face image of the user, the reconstructed face image from original PCA coefficients, and the four reconstructed face images from obtained PCA coefficients through L_2, L_1, LP and BIHT methods, respectively.

The first two set of images (Figures 4.6(a) and 4.6(b)) belong to two different users from the database and the reconstruction is carried out on biohashes with length of 1024-bits which are obtained from 200-dimensional PCA features. All four methods provide face images that look similar to the subject's original face image. Figures 4.7(a) and 4.7(b) illustrate the results for the same two users. The length of the biohashes used is 1024-bits, however, the only difference is the number of PCA coefficients used, which is 1024 instead of 200. It is immediately clear that estimating 1024-dimensional PCA features is harder than estimating 200-dimensional PCA features and the reconstructed face images show the difficulty in obtaining faces that are visually similar to the original face image. Among the four proposed methods, only the LP solution stands out for obtaining face images that look alike the original face of the subject. Figure 4.8 illustrates the reconstruction of face images using the LP method for various PCA feature vector dimensions and biohash bit lengths. It is clear that the reconstruction is visually more successful when the length of the PCA feature to be estimated is smaller and the biohash length is larger.

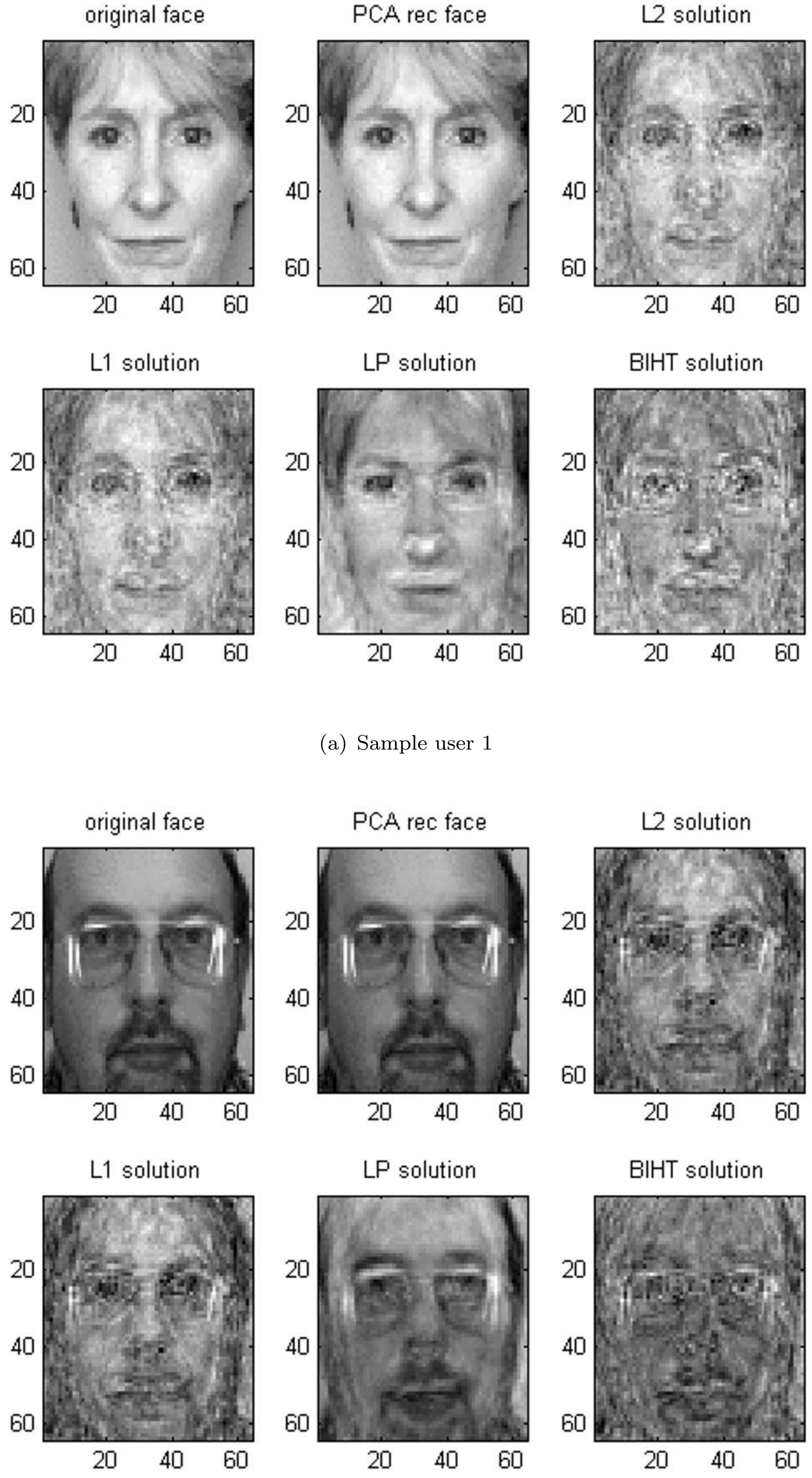
In addition to visually threatening the privacy of the system users, estimating feature vectors from biohashes might threaten their privacy in other biometric recognitions systems which use the same biometric characteristic (i.e., face information). To check whether reconstructed feature vectors are close to the original PCA feature vectors or not, we include face verification results of PCA vectors, (i.e., reconstructed feature vector is compared to corresponding original feature vector). The Euclidean distance is used to match two PCA vectors and each PCA vector is normalized in order to have zero mean and unit variance prior to comparison. The normalization step is required since the scale of the original PCA coefficients and the reconstructed ones might be different. We do not include all verification results for brevity, but the EERs for PCA-based face verification when 200-dimensional feature vectors are estimated from 1024-bits biohashes are given in Table 4.10 and the corresponding DET curves are shown in Figure 4.9.



(a) Sample user 1

(b) Sample user 2

Figure 4.6: Reconstructed face images from biohashes of length 1024-bits - PCA dimension 200



(a) Sample user 1

(b) Sample user 2

Figure 4.7: Reconstructed face images from biohashes of length 1024-bits - PCA dimension 1024

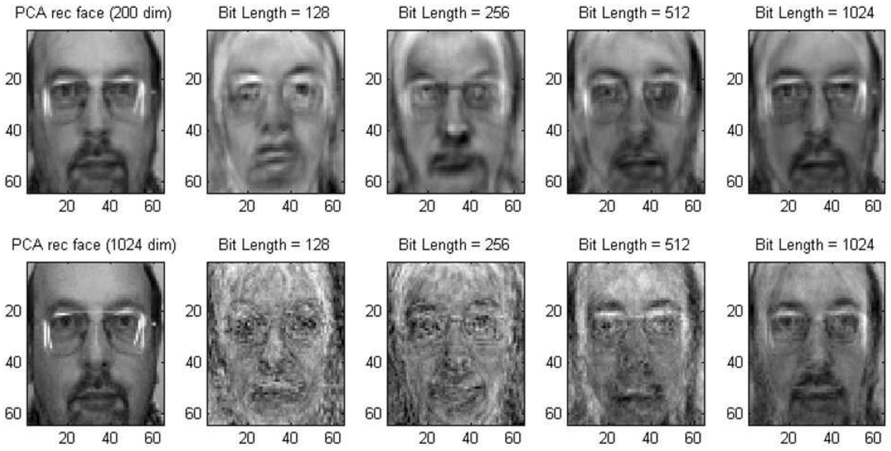


FIGURE 4.8: Reconstructed face images using the LP method for different biohash bit lengths (128, 256, 512 and 1024)

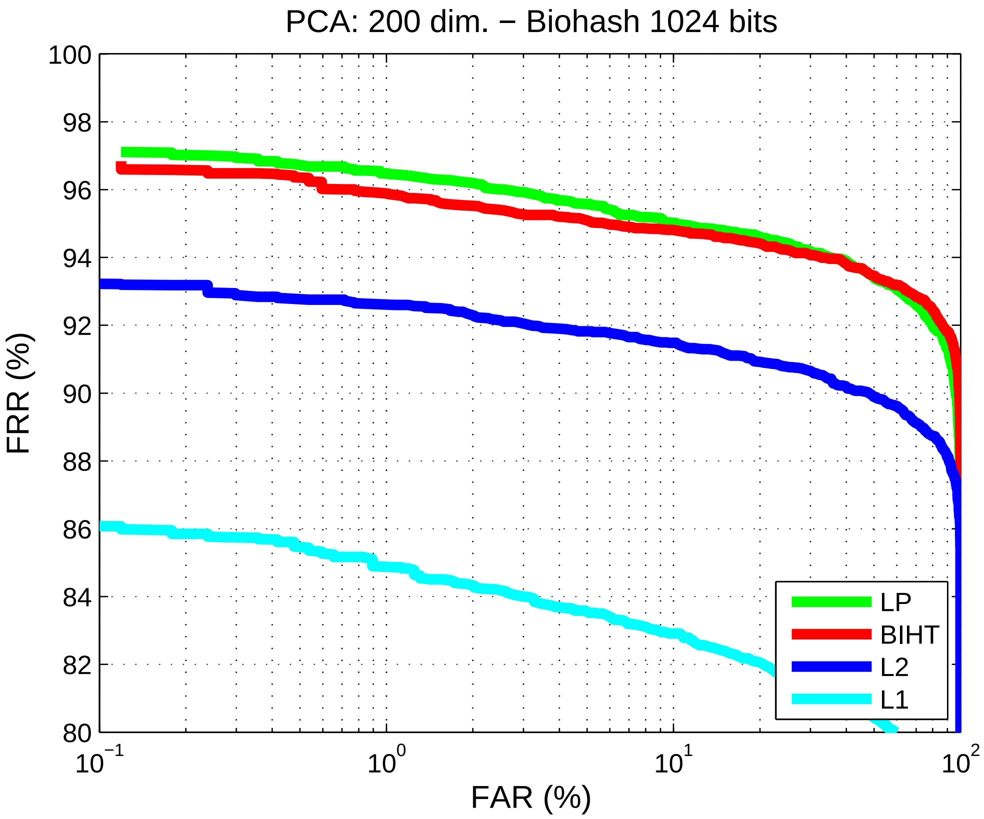


FIGURE 4.9: DET curves for direct feature level comparisons - 200-dimensional PCA feature vectors & biohash length = 1024-bits

TABLE 4.10: Equal Error Rates (%) for direct feature level comparisons - 200-dimensional PCA feature vectors & biohash length = 1024-bits

LP	91.161
BIHT	91.773
$\mathbf{L_2}$	88.338
$\mathbf{L_1}$	78.720

4.4 Discussion

Biometric template protection is a critical problem that needs to be addressed to enhance the public acceptance of biometric technologies and it is essential to develop a set of measures which can evaluate the strength of template protection techniques. Although biometric cryptosystems can be analyzed using information theoretical metrics such as entropy and mutual information, the suitability of theoretical analysis of the transformation-based methods is based on the hardness of the invertibility of the transformation.

When a user's biohash is obtained by an adversary, it can seriously undermine the security of the biometric system and privacy of users. If the secret key of a user is known to the adversary, the biometric feature of the user can be reconstructed from the user's biohash which might harm the subject's privacy and lead to illegitimate authentication to a system. Biometric hashing is claimed to be irreversible due to the random projection and quantization steps, however our study shows that an attacker is able to invert the transformed template to obtain a close approximation to the original biometric template.

This thesis proposes four novel ways to approximate the original biometric feature from the transformed template in a biometric hashing scheme and reveals security and privacy problems concerning the associated biometric system. We define three different attack scenarios under which we analyze the protection capability of biohashing. From the security point of view, these attacks enable an adversary to recover a biometric template under realistic assumptions and perform intrusion attacks to the biometric system. This study is the first to analyze the inversion of biohashes in one-bit compressive sensing framework. Experimental results show the superiority of this approach over minimum norm solutions. Biohashes that are created from feature vectors obtained by using LP and BIHT solutions to the one-bit compressive problem are equal to the original biohashes stored during enrollment and this is a serious threat to the security of the system. In addition, this study introduces rainbow attack in order to find a biometric template from a biometric database and use it to obtain a biohash that is same with or close to the original biohash of a subject.

Biometric hashing scheme is a generic template protection scheme that can be applied to various types of biometric features. In this paper, we focus on an orthogonal linear transform of face images, namely PCA (i.e., Eigenfaces). Several other studies on biohashing also use PCA ([9, 11]) or LDA ([14]) (i.e., Fisherface) which is another orthogonal linear transform that is invertible. Using the knowledge of the linear transform and its inversion, we analyze the privacy issues by reconstructing face images.

If the adversary knows system details (i.e., the PCA matrix, user's secret key, and other parameters), the obtained feature vectors can be used to reconstruct face images of the subject which is a direct threat to the privacy of system users. The quality of the reconstructed images depends on the number of bits and length of the original feature vector and the images illustrated in the last section visually confirm the success of the methods in reverting the biohash vectors. In this work, we study feature reconstruction and image reconstruction is carried out separately. Directly approximating images from biohash vectors may also be possible by integrating the PCA transformation with random projection matrix and solving the optimization problem by enforcing sparsity in the DCT or block-DCT domain. However, our initial experiments in this direction indicate that image level approximation approach lowers the performance both in security perspective (evaluated through EERs) and privacy perspective (evaluated through visual inspection of the reconstructed face images) due to the fact that the number of dimensions to be approximated is higher for images.

In the future, the effects of various improvements proposed for biometric hashing scheme might be investigated for security and privacy analysis by carrying out similar attacks on the improved versions of biometric hashing. In addition, weaknesses of the biometric hashing scheme should be explored and possible modifications should be introduced for better security and privacy protection capability in the light of the inversion attacks proposed in this study.

Chapter 5

Template Protection for Fingerprint Spectral Minutiae

Template protection options for fingerprint minutiae are discussed in section 2.4. It can be concluded that fuzzy vault has its own security and privacy drawbacks in addition to degradation in matching accuracy due to alignment issues and nonlinear distortion. For other alternative template protection schemes, a fixed-length feature vector or a binarized string is required. Spectral minutiae representation is one of the very few approaches that provides such a fixed-length representation for minutiae points.

In this chapter, we introduce spectral minutiae representation in detail and provide the first implementation of biometric hashing for spectral minutiae [40]. Our work in [40] is an initial attempt for protecting fingerprint minutiae templates. In the next chapter, we improve on the introduction to minutiae protection that we discuss here.

5.1 Biometric Hashing with Fingerprint Spectral Minutiae

Spectral minutiae representation [16] provides a fixed-length feature vector from minutiae location and direction information (i.e., (x, y, θ) in ISO 19794-2 standard format [60]). This fixed-length template for a fingerprint sample can then be combined with existing biometric template protection methods. In this work, biometric hashing scheme [7] is used for securing fingerprints and generating protected bit strings based on minutiae information only.

Biometric hashing transforms the biometric information of a user by using pseudo-random data which is generated from a user specified key or token. The combination of pseudo-random number and biometric data protects the biometric template against biometric fabrication when the corresponding token or knowledge of the randomization is not available to an adversary. Token-based randomization also enables revocation of one's biometric template via token replacement. This makes it possible to renew the biometric record of the subject in cases where it is compromised. Furthermore, different biohashes from the same fingerprint can be generated by using different tokens which allows a subject to be enrolled to two or more applications.

5.1.1 Spectral Minutiae Representation

The spectral minutiae representation of a minutiae set is a fixed-length feature vector that is invariant to translation, rotation and scaling [16]. These characteristics enable the combination of fingerprint recognition systems with template protection schemes and allow for fast minutiae-based matching. The spectral minutiae representation requires only minutiae information, therefore it is compatible with most of the existing fingerprint databases (in which only minutiae are saved and no additional information related to the finger, e.g., ridge count, singular points, and pores are kept) and minutiae-based fingerprint verification systems.

Complex spectral minutiae (SMC) is one of the three possible spectral minutiae representations proposed by Xu et al. [17] in order to obtain a fixed-length feature vector using minutiae location and orientation only. The other two alternatives are location based spectral minutiae (SML) and orientation based spectral minutiae (SMO) which are given in detail in [16].

In SMC, each minutiae is represented by a Dirac pulse and in order to reduce the sensitivity of minutiae locations to small variations in the spatial domain, a Gaussian low-pass filter is used to attenuate higher frequencies. This low-pass filtering in the frequency domain corresponds to a convolution in the spatial domain where every minutia is now represented by an isotropic two-dimensional Gaussian function with standard deviation σ_C. Minutiae locations on a fingerprint image and the corresponding Gaussian functions are illustrated in Figure 5.1. Minutiae orientation is incorporated into this representation by assigning each Gaussian a complex amplitude $e^{j\theta_i}$, where θ_i is the orientation of

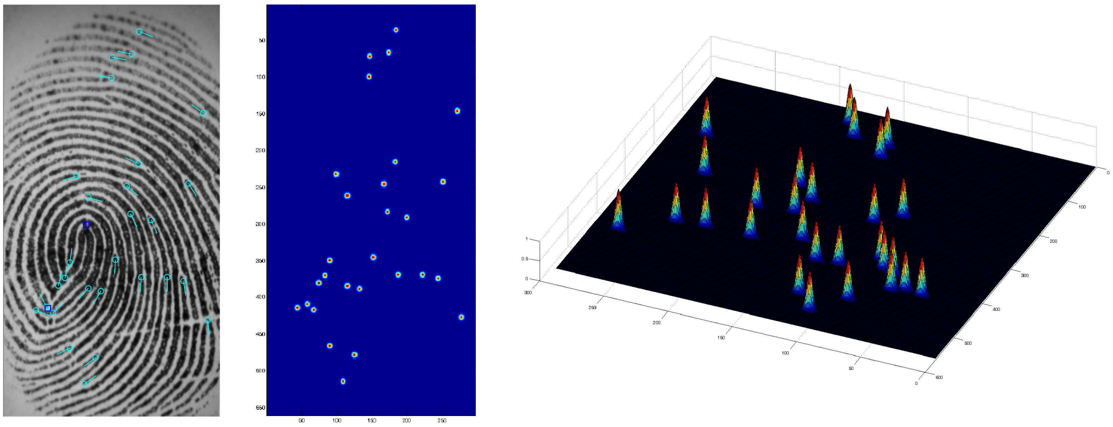


FIGURE 5.1: Minutiae locations and set of minutiae represented by Gaussian functions

the corresponding minutia. For a set of Z minutiae with locations $(x_i, y_i)|_{i=1}^{Z}$, the complex spectral representation, $\mathbf{M_C}(w_x, w_y; \sigma_C^2)$, is obtained by evaluating the magnitude of the Fourier spectrum (5.1) on a polar-logarithmic grid as:

$$\mathbf{M_C}(w_x, w_y; \sigma_C^2) = \left| \exp\left(-\frac{w_x^2 + w_y^2}{2\sigma_C^{-2}} \right) \sum_{i=1}^{Z} \exp(-j(w_x x_i + w_y y_i) + j\theta_i) \right|, \qquad (5.1)$$

where w_x and w_y are the spatial frequencies in the x and y directions.

The Fourier spectral magnitude is mapped onto a polar-logarithmic coordinate system as $\lambda = \sqrt{w_x^2 + w_y^2}$ and $\beta = \arctan(w_y/w_x)$, where λ corresponds to the radial direction and β corresponds to the angular direction. In the radial direction $M = 128$ samples are used between $\lambda_l = 0.05$ and $\lambda_h = 0.63$. In the angular direction $N = 256$ samples are used between $\beta = 0$ and $\beta = 2\pi$. The resulting complex spectral representation of a minutiae set is a 128×256 matrix (Figure 5.2).

5.1.2 Protecting SMC Template with Biometric Hashing

Biometric hashing, initially applied to FingerCode [32] fingerprint templates by Jin et al. [7], is a two factor authentication approach that combines a fingerprint with a user specified key/token and generates a unique compact code per person (Figure 5.3). A bit string from biometric data is created by taking the inner product of a fixed-length fingerprint feature vector and the pseudo-random number sequence that is generated

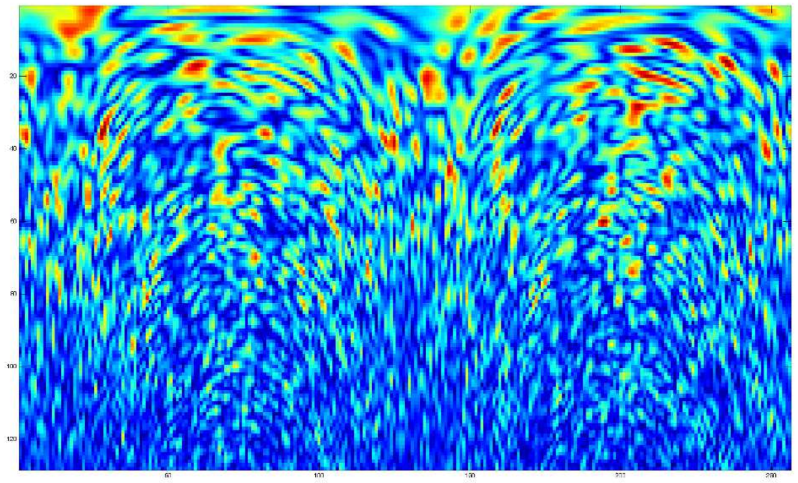


FIGURE 5.2: Complex spectral minutiae representation of a fingerprint

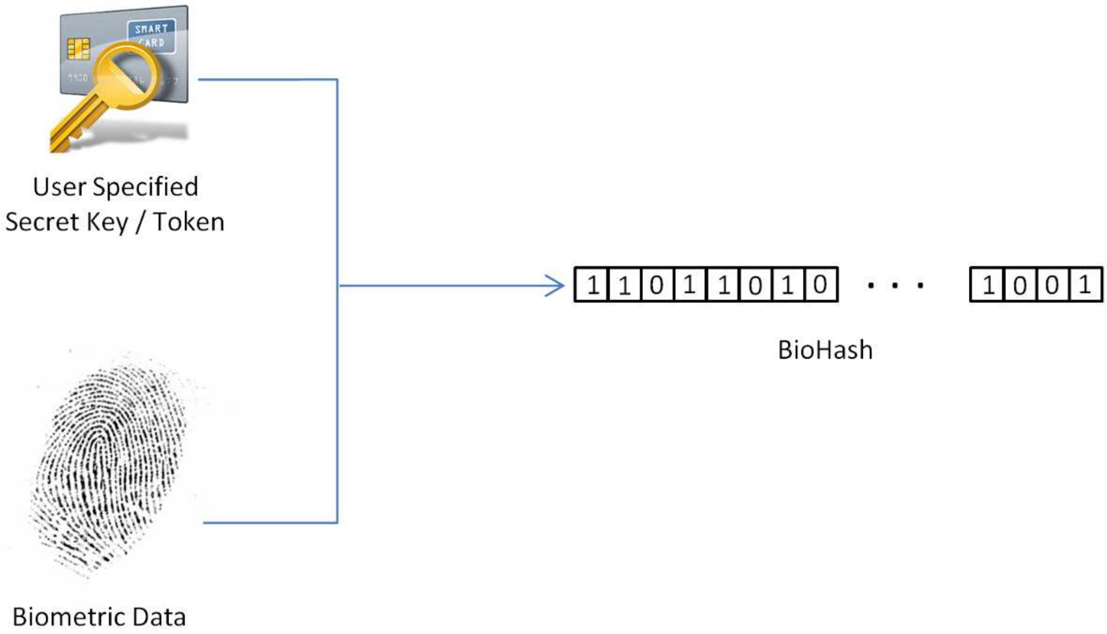


FIGURE 5.3: Two factor authentication - secret key and fingerprint

using the key as the seed. Each bit is decided based on the sign of the result by comparing to a pre-defined threshold.

In our study, we convert the $M \times N$ spectral fingerprint feature ($\mathbf{M_C}$) to a bit string, $\mathbf{b} \in \{0,1\}^p$, by applying the biohashing scheme to complex spectral minutiae features. Each column of $\mathbf{M_C}$ is a M-dimensional column vector. Randomly projecting each column of $\mathbf{M_C}$ to k dimensions and then thresholding the resulting vector, we obtain

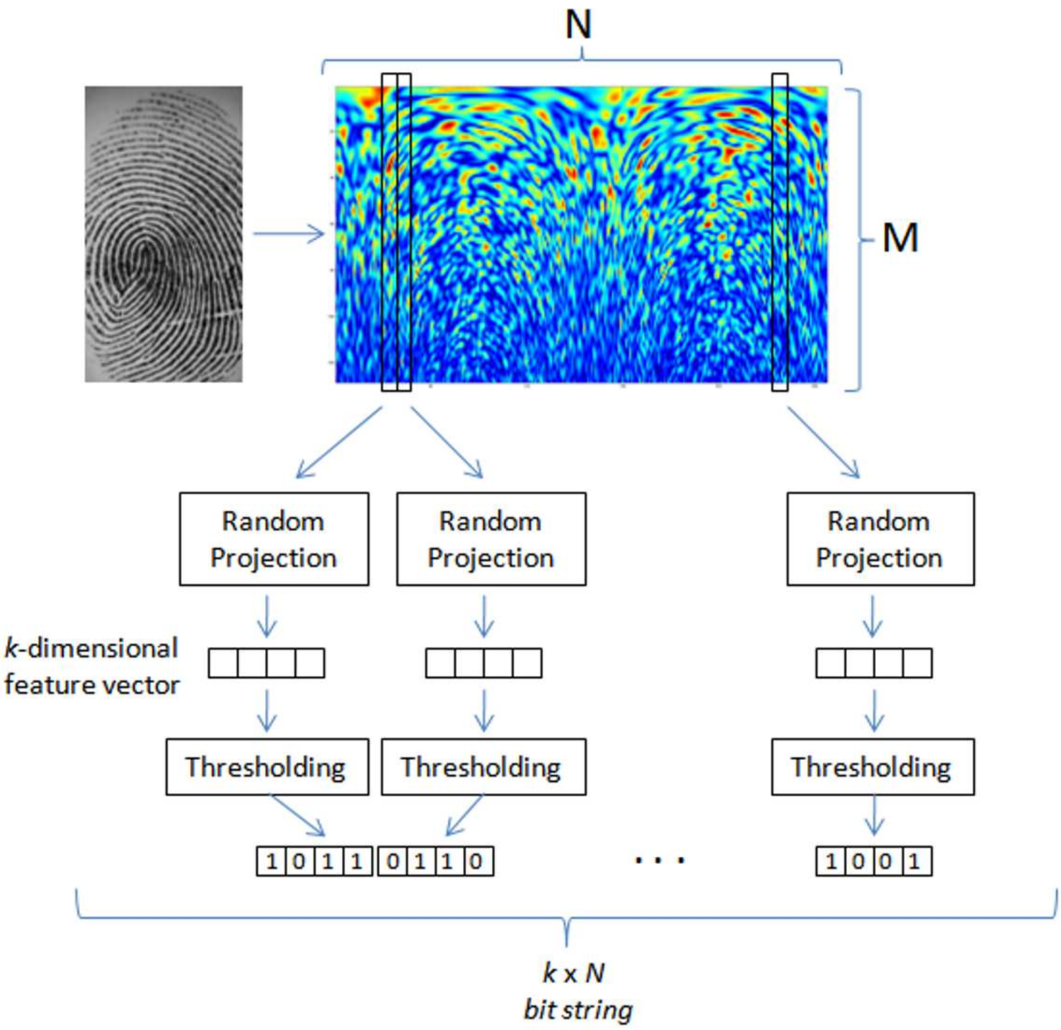


FIGURE 5.4: BioHashing procedure - from spectral representation to bit string

a k-length bit string per column. The mean value of the k-dimensional feature vector is used as the quantization threshold. We apply the same procedure to each column of $\mathbf{M_C}$ and concatenate the bit strings to create a p-length bit string, where $p = k \times N$.

In our implementation, the spectral fingerprint representation that we create from a minutiae set ($\mathbf{M_C}$), is a 128×256 matrix. Each column $\mathbf{f}_n$ of this matrix is a 128-dimensional column vector and it is reduced to k dimensions by multiplying it ($\mathbf{R} \cdot \mathbf{f_n}$) with the random projection matrix, $\mathbf{R}$ (which is a $k \times 128$ matrix). Thresholding the resulting k-dimensional feature vector by using its mean value as the threshold, we obtain a k-length bit string. The outputs of each column of $\mathbf{M_C}$ are then concatenated in order to create a bit string of length $k \times 256$. We evaluate different values of k and use $k = 4$

resulting in a 1024-bit final feature vector to obtain a high verification accuracy while keeping the feature vector small (Figure 5.4).

5.1.3 Experiments and Results

5.1.3.1 Experimental Settings

We evaluate our algorithm on publicly available FVC2002 fingerprint databases, namely DB1A, DB2A, and DB3A [61]. DB1 and DB2 consist of fingerprint images captured with optical sensors whereas images in DB3 are captured with a capacitive fingerprint sensor. We select these three databases in order to evaluate the performance of our method for different image capturing technologies and leave the synthetic fingerprint database DB4 out in the experiments. For performance evaluations, we adopt the equal error rate (EER), which is the error rate when the frequency of false accepts (FAR) and the frequency of false rejections (FRR) are equal to each other.

The minutiae sets are obtained by a commercial automatic minutiae extractor (Verifinger 4.4 SDK). We use our algorithm in a high security scenario as suggested in the original spectral minutiae work [16]. In FVC2002 databases, some of the samples are obtained by requesting users to provide fingerprints with exaggerated displacement and rotation. In a high security scenario, users are aware that cooperation is crucial for security reasons. Therefore, only four out of eight samples are chosen for each subject (1-2-6-7 for DB1, 1-2-7-8 for DB2, and 1-2-6-7 for DB3). Following the verification setting described in FVC competitions, we use all possible combinations to match genuine pairs and the first sample of each subject is chosen for imposter matches. Without making symmetric comparisons, this results in a total of 600 $(4 \times 3 \times 100/2)$ genuine matches and 4950 $(99 \times 100/2)$ imposter matches for 100 subjects.

5.1.3.2 Results for the Naive Model

We evaluate the spectral minutiae representation and biohash of spectral minutiae representation on three databases. For comparison, we also include the results from two other matching methods: i) matching two fingerprints based on the correlation of their complex spectral minutiae (called SMC-Correlation) and ii) matching two fingerprints using a minutiae-based commercial matcher which is also used for minutiae extraction.

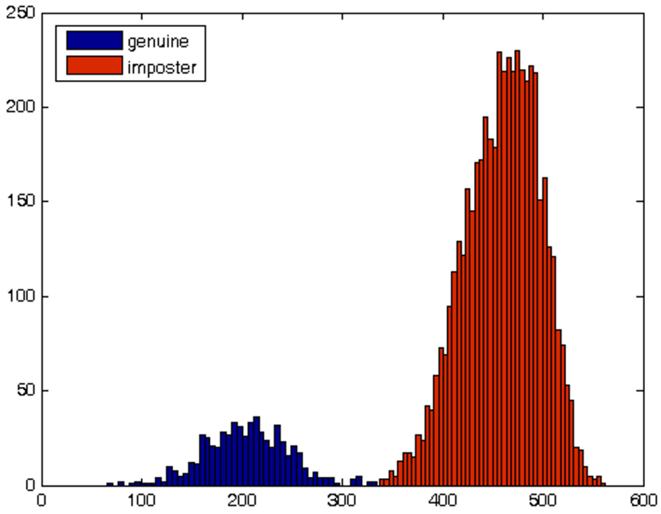


Figure 5.5: Genuine and imposter distance distribution for the FVC2002DB1A database

The equal error rates (EER) for all methods on three databases are given in Table 5.1. As can be seen in this table, we obtain a 0% EER for all databases when biohashing is applied on the spectral minutiae features.

Table 5.1: EER on FVC2002 databases

	SMC-Correlation	**Minutiae Matching**	**SMC-BioHash**
FVC2002 DB1	6.50%	0.50%	0.00%
FVC2002 DB2	6.47%	0.83%	0.00%
FVC2002 DB3	11.68%	2.50%	0.00%

Two factor biometric hashing (fingerprint + user specified tokens) provides a clean separation of genuine and imposter populations along with a zero EER level. As an example, genuine and imposter distance distribution for the FVC2002 DB1 database is illustrated in Figure 5.5. It is observed that the highest genuine distance is smaller than the lowest imposter distance, therefore a perfect separation between genuine and imposter distances is obtained.

5.1.3.3 Results for Stolen Key Scenario

We also evaluate the performance of our proposed scheme on a stolen key scenario, where an unauthorized imposter acquires the secret key/token of a genuine user but

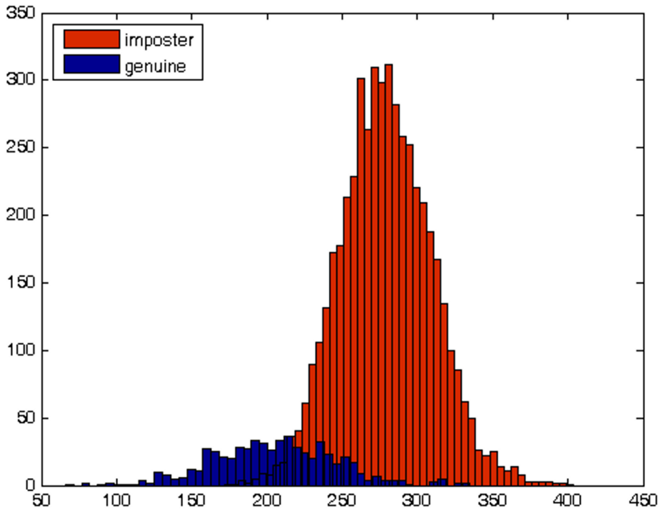


Figure 5.6: Genuine and imposter distance distribution for the FVC2002DB1A database for the stolen key scenario

does not have the claimed person's fingerprint information. In this case, the imposter sends his/her fingerprint template and the secret key/token of the genuine user in order to be authenticated as the genuine user. This is a serious threat to the system as the pseudo-random vectors generated using the secret key has a considerable influence on the generated bit string, therefore on the matching score.

Assuming that the key is unknown at all times (never stolen) makes the use of biometric unnecessary for real authentication scenarios. In order to analyze the effect of the key/token and generated random vectors on the resulting bit strings, we conduct experiments in a stolen key scenario where an imposter attempt has the same secret key with the user that he/she is intended to authenticate as. The equal error rates for this scenario are given in Table 5.2.

Table 5.2: EER on FVC2002 databases - stolen key scenario

	SMC-Correlation	**SMC-BioHash (Stolen Key)**
FVC2002 DB1	6.50%	14.77%
FVC2002 DB2	6.47%	13.10%
FVC2002 DB3	11.68%	26.46%

While these error rates are considerably high in this case, they are in the same range as other results obtained with fingerprint biohash implementations. For instance, the straightforward biohash implementation that uses the FingerCode [32] reported in [45] achieves a 15%, 15%, and 27% EER on FVC2002 DB1-DB3 databases respectively for the stolen key scenario (the BASE row in Table 5 of [45]). Our error rates for this case are slightly better in the same scenario. The same authors reported improved results (7%, 6.8%, and 22% on FVC2002 DB1-DB3 respectively) with a classifier combination approach that aims to reduce the stability issues of biohash, presumably with a system significantly slower and larger than ours.

5.1.3.4 Analysis

Two factor biometric hashing scheme used in this study improves the verification accuracy of biometrics alone and provides a clear separation of the genuine and imposter distances achieving a zero EER level. With this method, a unique compact code per person is obtained which is easy to match via bit-wise XOR operation (Hamming distance).

Our main contribution is providing the first implementation of the biohashing scheme with spectral minutiae representation. The proposed scheme is computationally fast as it only uses column-wise random projection of the spectral minutiae matrix while achieving a 0% EER in the 1-to-1 verification scenario.

The original spectral minutiae features are 8096-dimensional (128 × 256). In order to create a 1024-bits string, one needs to generate a random projection matrix of size 1024 × 8096. Instead, we propose to use a single 4 × 128 random projection matrix to multiply with each column of SMC (128-dimensional column vectors). This results in a computationally low random projection operation as well as an adaptive thresholding for each column of SMC, instead of generating a larger projection matrix (which takes more time to generate for higher number of vectors - 1024 instead of 4) and using a single threshold for quantization.

5.2 Discussion

Biometrics is a key factor for human identification or identity management since it bases recognition task on intrinsic human characteristics and the person to be authenticated should be physically present. However, biometrics suffer from high false rejection of valid users when a high security scenario is desired with a low false acceptance rate.

In this study, we propose a biometric hashing approach for fingerprint identification based on minutiae information. Using the spectral minutiae representation of a fingerprint minutiae set, we create a fixed-length bit string by randomly projecting spectral minutiae feature vectors. With this approach, one can obtain perfect separation between the genuine and imposter population and the system provides a 0% equal error rate, which is desired for all identity verification systems. In addition, in case the secret key of a valid user is stolen, the system allows acceptable error rates for imposter authentication attempts with a valid secret key. Also, biometric revocation becomes feasible through secret key (token) replacement, which addresses the cancellability issue.

Possible future work in this direction includes different quantization methods following the random projection of spectral minutiae feature vectors in order to provide non-invertibility of the protected template. The processing time for comparing two bit strings is very low as Hamming distance metric is used for bit string comparison. However, extracting the feature vector from spectral minutiae representation takes considerable time. This is another issue which should be addressed. It is also important to investigate the stolen key scenario and decrease the error rates when the secret key of a user is stolen by creating intelligent projections.

This study is an initial attempt for protecting fingerprint minutiae templates. However, it should be noted that a large overlapping area between fingerprints is required for spectral minutiae representation to perform well. Missing or spurious minutia leads to decreased verification performance. In the remaining chapters of this thesis, we seek for a better fixed-length representation for minutiae.

Chapter 6

GMM-SVM Fingerprint Verification

In this chapter, our ultimate goal is to describe an underlying framework that enables the generation of a fixed-length feature vector representation for fingerprint minutiae. The framework draws upon the work of Campbell et al. on support vector machines using GMM supervectors for speaker verification [18]. Each minutia and its neighbors within a specified radius are represented as a 2D image by placing two-dimensional Gaussians at the locations of neighbor minutiae. DCT coefficients of this patch image are rearranged based on zig-zag scanning and the first D DCT coefficients of this patch image are used to represent each minutia as a D-dimensional feature vector. A single user-independent GMM universal background model (UBM) is trained from a collection of fingerprints to represent the distribution of DCT features. A fingerprint is then represented with its probabilistic alignment into the UBM mixture components and a GMM supervector is created from the stacked first order statistics of the mixture components.

For a given enrollment fingerprint sample, a two-class linear SVM is trained in order to create a model template that discriminates positive samples from negative samples. The matching between a query fingerprint and the model template is performed by computing a single inner product between the target fingerprint SVM model and the query fingerprint GMM supervector. Next, the GMM-SVM features are binarized using asymmetric locality sensitive hashing. The overall GMM-SVM fingerprint verification system is illustrated in Figure 6.1. The performance of our framework is evaluated in a

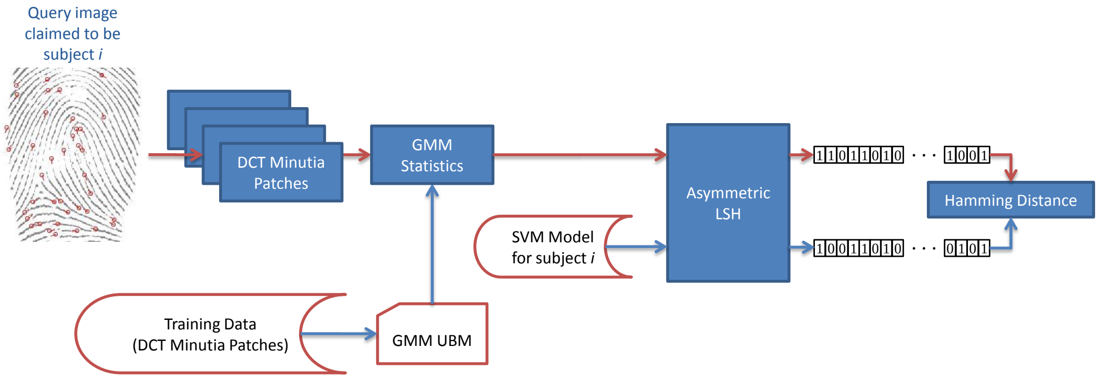


Figure 6.1: Overall framework of the GMM-SVM fingerprint verification system

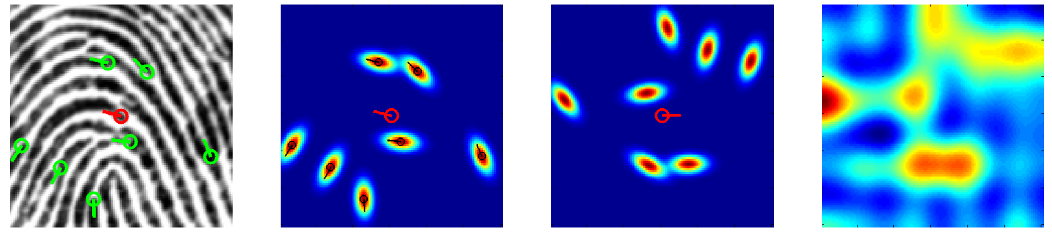
one-to-one fingerprint verification setup and the results on the FVC2002DB1A and the FVC2002DB2A databases demonstrate that our approach performs better than existing fixed-length methods.

6.1 DCT-based Minutiae Patch Representation

6.1.1 Minutia Patch

A minutia patch is a local representation that encodes a minutia and its geometric relations with other minutiae that are closely located. Each minutia patch consists of a central minutia m_c and its neighboring minutiae within a radius R (Figure 6.2(a)). In order to directly compare two minutia patches, without any registration for the relative alignment of fingerprints, a relatively invariant representation using m_c as a reference is required. The central minutia m_c can be used to define a new coordinate system where its position would be the center of the system and its orientation would give the direction of the x-axis. In this new coordinate system, the coordinates and orientations of the neighbor points would be translated and rotated accordingly. This representation scheme is inspired from minutiae vicinities described in [38].

In this representation, a global set of minutiae is converted into a collection of several local minutiae sets and a patch is constructed for each minutia of a fingerprint. This also enables two fingerprints to be matched by locally comparing patches pairwise and calculating their similarity score using the local scores of the best pairs. Although global coherency in the minutiae set is not utilized, the local approach has the advantage of



(a) A minutia and its neighbors in R

(b) Gaussian representation for neighbor minutiae

(c) Neighbor minutiae at the new coordinate system

(d) Reconstructed minutiae patch image from DCT coefficients

Figure 6.2: DCT representation of a minutia patch image

limiting the crucial elastic distortion problem in fingerprint matching. In the local area of a patch, distortion due to the elasticity of the skin is negligible. The radius used in the local approach is of great importance. The neighborhood of the central minutia should contain several minutiae in order to be sufficiently discriminative but at the same time it should be small enough to be considered as a local area.

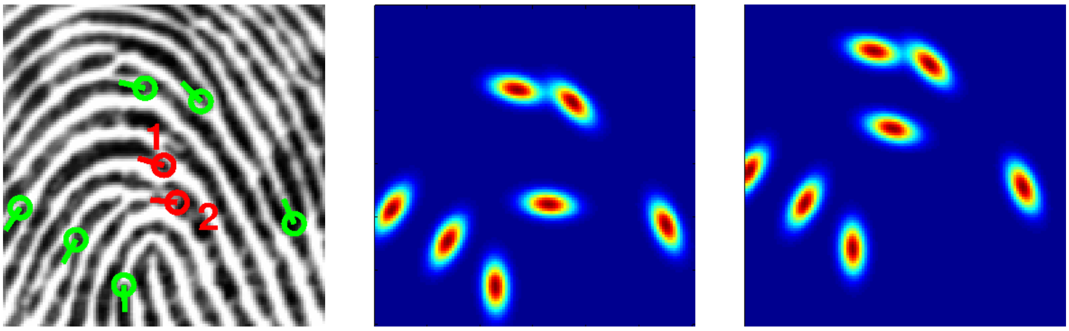
6.1.2 Gaussian minutia patch image

Within a specific radius R, the number of neighbor points of a central minutia varies and this leads to a minutia representation of unknown length. In order to obtain a fixed length representation, one can use a rectangular grid of size $(2R+1) \times (2R+1)$ where the central minutiae is at the center. Each neighbor minutia is then inserted into this grid with respect to its relative location to m_c on the fingerprint.

Representing a minutia with a single point in the spatial domain increases the sensitivity of minutiae positions to small variations and does not maintain direction information. Instead, each neighbor minutia is represented by a two-dimensional multivariate Gaussian. We first consider a template anisotropic Gaussian:

$$f(\mathbf{x}, \mu, \Sigma) = \frac{1}{2\pi\sigma_1\sigma_2} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)}, \tag{6.1}$$

where $\Sigma = diag(\sigma_1, \sigma_2)$ is the covariance matrix with $\sigma_1 > \sigma_2$. A Gaussian is centered at each neighbor minutia location and its covariance matrix is selected such that the major axis coincides with the minutia orientation as illustrated in Figure 6.2(b). For a neighbor minutia with relative position (x_i, y_i) and relative angle θ_i as compared



(a) Two neighbor minutiae (b) Minutia patch for minutia 1 (c) Minutia patch for minutia 2

Figure 6.3: Selected minutiae patches of two neighbor minutia from the same fingerprint image (before rotation)

with the central minutia in the patch, the template Gaussian is translated to (x_i, y_i) and rotated with θ_i. This makes the mean of Gaussian $\mu_i = [x_i, y_i]^T$ and covariance matrix $\Sigma_i = R(\theta_i)\Sigma R(\theta_i)^T$, where $R(\theta_i)$ is a rotation matrix [1]. The patch image is then generated as a sum over these shifted Gaussians:

$$I(\mathbf{x}) = \sum_{i=1}^{N_p} f(\mathbf{x}, \mu_i, \Sigma_i) \tag{6.2}$$

where N_p is the number of neighbor minutiae. Sample minutia patch images selected from a fingerprint are illustrated in Figures 6.3(b) and 6.3(c). Please note that, the central minutia is not directly included in this representation, but it defines the new coordinate system and the neighbors of the patch.

6.1.3 DCT representation for minutia patches

Although minutia patch images capture the required information for fingerprint matching, $(2R+1)^2$-dimensional representation for each minutia brings heavy computation and storage requirements. Discrete cosine transform (DCT) is often used in image processing, especially for lossy compression (e.g., JPEG), due to its strong energy compaction property. It expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. Since most of the signal information tend to be concentrated in a few low-frequency components of the DCT, discarding small high-frequency components results in compact representation of the signal. By keeping only

[1] Please note that, Gaussians are placed prior to the rotation with respect to the orientation of the central minutia θ_c.

the first D 2D-DCT coefficients after performing zig-zag scanning, each minutia patch image is represented as a D-dimensional feature vector. This enables two patches to be easily compared via Euclidean distance between their D-dimensional feature vectors.

Minutiae Pair Matching via DCT Patches We conduct an evaluation to assess the discriminative power of our DCT minutia patch representation. To compare two fingerprints, fp_1 and fp_2, a pairing matrix that contains similarity scores between patches of fp_1 and patches of fp_2 is constructed. The scores are computed using a decreasing function that converts the Euclidean distances between DCT coefficients to a score (i.e., $g(x) = 1/(1 + e^{x/\tau})$).

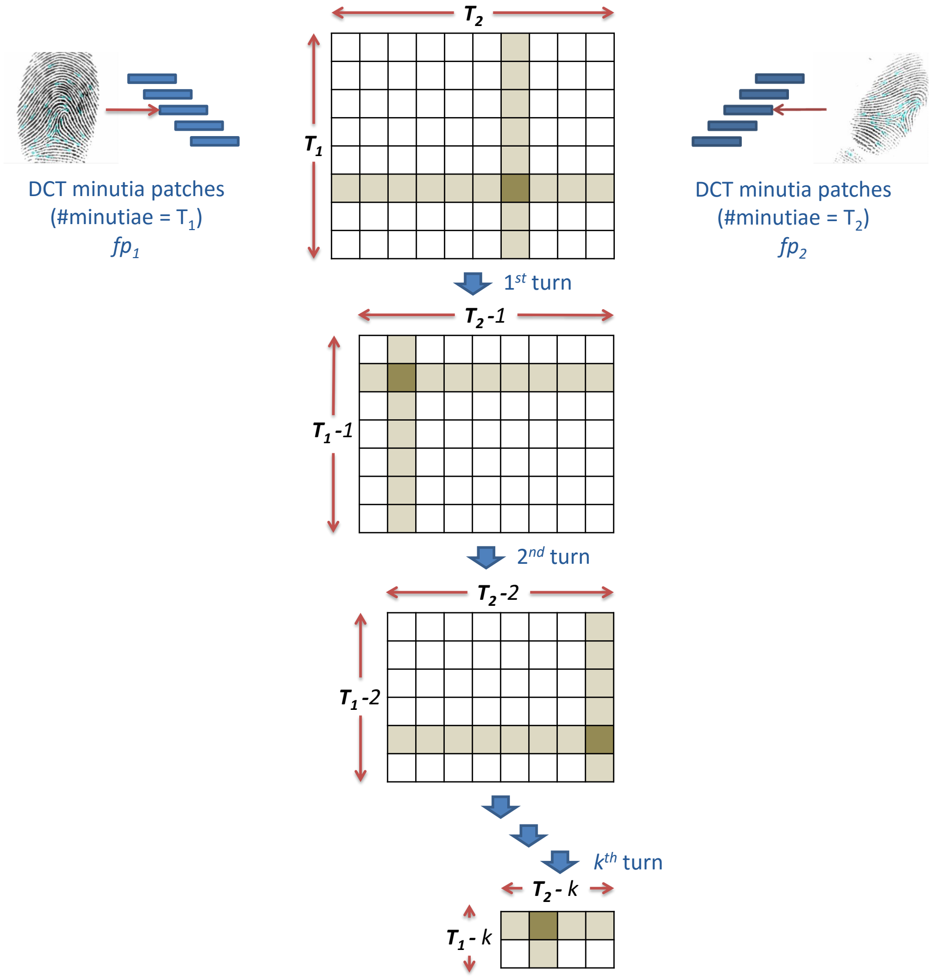


FIGURE 6.4: Minutiae pairing matrix and selection of highest score at each turn

A closest neighbor search algorithm is applied to the pairing matrix in order to select the best association of minutiae. At each turn, a minutiae pair from fp_1 and fp_2 with the highest matching score is identified and removed from the matrix (Figure 6.4). The final score between two fingerprints is computed by accumulating the matching scores of identified pairs during the search.

In the evaluation, the FVC2002DB1A database [61] which has 8 impressions of 100 different fingerprints captured with an optical sensor is used. Following the performance evaluation protocol of FVC2002 [62], 2800 genuine and 4950 imposter comparisons are performed. An Equal Error Rate (EER) of 4.46% is achieved for $D = 50$. Although, the achieved EER is worse than the state of the art [61, 62], it arguably confirms the discriminative capability of minutia patches.

6.2 GMM Supervector Training

Gaussian mixture models (GMMs) have been dominantly used for modeling in text-independent speaker verification. The distribution of features extracted from speech segments (i.e., frames of an utterance) is modeled by performing background model adaptation of GMMs. First, a universal background model (UBM) is trained from set of frames and then the speaker model for the i^{th} speaker is derived by adapting the universal background model to match the observations of the speaker. Recently, the use of GMM for modeling feature distribution has also become an effective approach for face verification [63].

Similar to the frames of a speech utterance or the blocks of a face image, minutiae points of a fingerprint are separate observations of the same underlying signal. DCT patch representation of minutiae is used to train a universal background minutiae model. The UBM is a large GMM trained to represent the distribution of features. From a huge database of fingerprints, a large number of minutiae patches are extracted as training data and they are pooled to train the UBM via EM (expectation maximization [64]) algorithm[2]:

[2] For further details, please refer to [65].

$$g(\mathbf{x}) = \sum_{i=1}^{N} w_i p_i(\mathbf{x}) \tag{6.3}$$

where w_i are the mixture weights and $p_i(\mathbf{x})$ is the unimodal Gaussian density with mean $\mathbf{m}_i$ and covariance $\mathbf{\Sigma}_i$ (diagonal covariance is assumed as this requires fewer observations to train from).

Given a fingerprint with T minutiae, $\mathbf{x}_t, t = 1, ..., T$ are the DCT minutia patches for each minutia. The estimates of first order statistics for the fingerprint data are computed for mixture i in the UBM as:

$$E_i(\mathbf{x}) = \sum_{t=1}^{T} Pr(i|\mathbf{x}_t)(\mathbf{x}_t - \mu_\mathbf{i}) \tag{6.4}$$

$$Pr(i|\mathbf{x}_t) = \frac{w_i p_i(\mathbf{x}_t)}{\sum_{j=1}^{M} w_j p_j(\mathbf{x}_t)} \tag{6.5}$$

Using only the first order statistics $(E_i(\mathbf{x}))$, a GMM supervector is formed by concatenating the first order statistics of each mixture. The GMM supervector maps a fingerprint to a high-dimensional vector of size $DN \times 1$, where D is the number of DCT coefficients and N is the number of Gaussians in the mixture (Figure 6.5). Please note

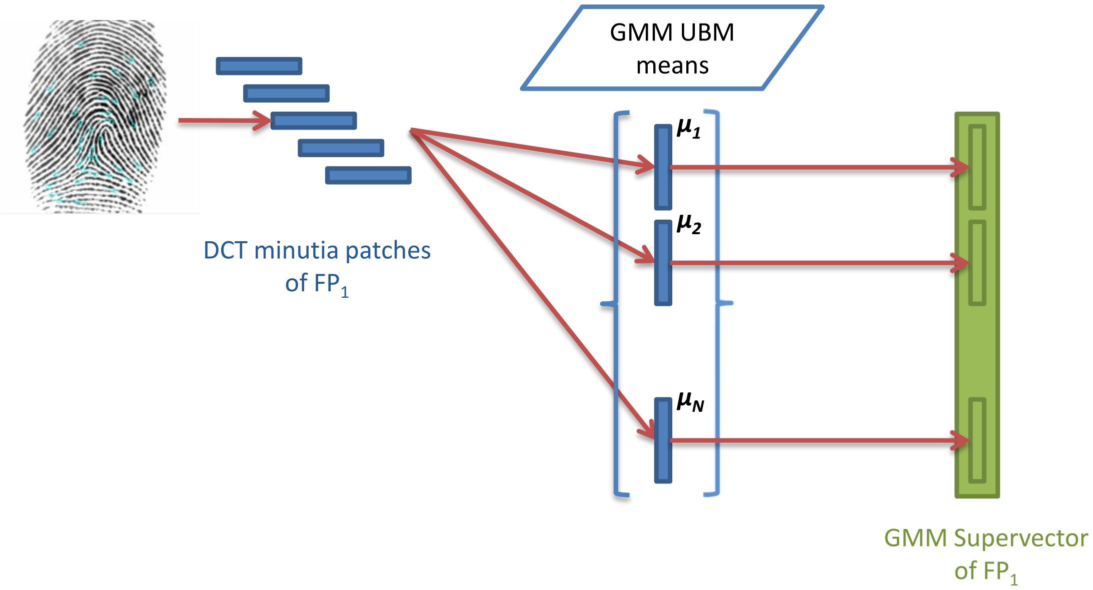


Figure 6.5: GMM supervector generation from a single fingerprint

that, we do not perform MAP adaptation as done in [18, 63, 65, 66] for adapting a speaker model. Our experiments shows that using first order statistics without MAP adaptation performed better, so we employ first order statistics only.

6.3 Linear SVM Training for Template Generation

An SVM is a two-class linear classifier constructed from sums of a kernel function K

$$f(\mathbf{x}) = \sum_{i=1}^{L} \alpha_i t_i K(\mathbf{x}, \mathbf{x}_i) + d \tag{6.6}$$

where t_i are the ideal outputs (either 1 or -1), d is a learned constant, $\sum_{i=1}^{L} \alpha_i t_i = 0$, and $\alpha_i > 0$. The vectors $\mathbf{x}_i$ are support vectors and obtained from the training set by an optimization process. The kernel K is constrained to have certain properties so that K can be expressed as an inner product, $K(\mathbf{x}, \mathbf{x}_i) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})$, where $\Phi(\cdot)$ is a mapping to a higher dimension.

SVM provides a suitable solution to fingerprint verification problem, since it is fundamentally a two-class problem. We aim to decide whether the fingerprint comes from the user or the fingerprint belongs to someone else. As the number of features is large in our problem (DN), we do not need to map data to a higher dimensional space and use linear kernel (i.e., $K(\mathbf{x}, \mathbf{x}_i) = \mathbf{x}_i^T \mathbf{x}$. In practice, the linear kernel tends to perform very well when the number of features is large. In addition, GMM supervector has already been employed as a linear kernel with a simple diagonal scaling [18, 67]. The SVM in (6.6) can be expressed as:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{L} \alpha_i t_i \mathbf{x}_i^T \mathbf{x} + d \\ &= \left(\sum_{i=1}^{L} \alpha_i t_i \mathbf{x}_i \right)^T \mathbf{x} + d = \mathbf{w}^T \mathbf{x} + d \end{aligned} \tag{6.7}$$

which reduces two-class classification to an inner product between the classifier model $\mathbf{w}$ and GMM supervector $\mathbf{x}$. The model $\mathbf{w}$ is solved by minimizing:

$$\min_{\mathbf{w},d} \left(\frac{1}{2} \|\mathbf{w}\|_2 + C \sum_i H_1 \left(t_i(\mathbf{w}^T \mathbf{x}_i + d) \right) \right) \tag{6.8}$$

where $H_1(z) = \max(0, 1 - z)$ is the "Hinge Loss" and C is a regularization parameter that controls a tradeoff between a low error on the training data and the ability to generalize well.

We use SVM to create a model $\mathbf{w}$ (which we also refer to as a reference template) for an enrollment fingerprint sample $\mathbf{f}_{enroll}$. This is achieved by training an SVM using the GMM supervector of $\mathbf{f}_{enroll}$ as a positive sample (labeled as +1) and GMM supervectors of fingerprints from example imposters as negative samples (labeled as -1). Given a query fingerprint sample $\mathbf{f}_{query}$, its matching score for the subject i is the inner product between $\mathbf{w}_i$ and $\mathbf{x}_{query}$, where $\mathbf{w}_i$ is the SVM classifier model for the subject i and $\mathbf{x}_{query}$ is the GMM supervector of $\mathbf{f}_{query}$. The verification decision is based upon whether the score $\mathbf{w}_i^T \mathbf{x}_{query}$ is above or below a threshold. This approach provides one-to-one fingerprint matching as only one single training sample for each class is used to train the template model. It corresponds to comparing a gallery fingerprint to a query fingerprint as done in all other fingerprint verification systems.

6.3.1 Initial Experiments and Results

We perform one-to-one fingerprint verification experiments on the FVC2002DB1A fingerprint database [61]. For minutiae extraction, a commercial fingerprint minutiae extractor (which participated in FVC-onGoing [68], Ongoing MINEX [69] and FpVTE 2012 [70]) is used to obtain minutiae information in ISO 19794-2 format (x, y, θ) [60].

In order to create patches for each minutia, all neighbor minutiae within a radius $R = 60$ pixels at 500 dpi resolution are used. This results in minutia patch images of size 121×121 pixels. For DCT representation of patches, the first 50 DCT coefficients after zig-zag scanning are kept (i.e. $D = 50$), which means that a minutia is represented along with its local information via a feature vector of length 50.

We use 15808[3] fingerprints from publicly available FVC databases and an in-house fingerprint database collected via an optical reader. The details of these databases (number

[3] 32 minutiae in FVC2006DB1 have zero neighbors within R, therefore they are not used in GMM training.

DB Name	**#Fingers**	**#Fingers × #Samples/Finger**
FVC2002DB2	800	100 × 8
FVC2002DB3	800	100 × 8
FVC2002DB4	800	100 × 8
FVC2004DB1	800	100 × 8
FVC2004DB2	800	100 × 8
FVC2004DB3	800	100 × 8
FVC2004DB4	800	100 × 8
FVC2006DB1	1648	140 × 12
FVC2006DB2	1680	140 × 12
FVC2006DB3	1680	140 × 12
FVC2006DB4	1680	140 × 12
IN-HOUSE DB	3520	440 × 8
TOTAL	**15808**	

Table 6.1: Number of fingerprints used in GMM training

of fingers and samples per finger) can be found in Table 6.1. Our target database, FVC2002DB1A, is not included in GMM training to prevent any bias that might favor supervector representation in the advantage of the FVC2002DB1A database. The GMMs are trained for different number of Gaussians (1024, 2048, and 4096) and their results are reported separately. Once the universal models are trained, we extract first order statistics of fingerprint samples from FVC2002DB1A and produce supervectors for GMMs with different number of Gaussians, which results in supervectors of dimensions 51200 (1024×50), 102400 (2048×50), and 204800 (4096×50).

For the enrollment of target fingerprints, we train an SVM for each fingerprint sample using the target GMM supervector and the set of imposter GMM supervectors labeled as -1, using the first impression of each subject as imposters. The weight vector of the SVM classifier model is the template for the enrolled fingerprint sample. During verification, GMM supervector of the query fingerprint is compared to the template of the claimed identity and their inner product is used to give an accept or reject decision.

The verification protocol is as follows:

i) Each impression is matched against the remaining impressions of the same finger. The total number of genuine tests is 5600 ($8 \times 7 \times 100$).

ii) The first impression of each finger is matched against the first impression of the remaining fingers. The total number of imposter tests is 9900 (99×100).

# Gaussians	EER
1024	2.191%
2048	1.911%
4096	1.633%

Table 6.2: Equal error rates for GMMs with different number of Gaussians

Since $\mathbf{w}_{fp_1}^T \mathbf{x}_{fp_2}$ is different from $\mathbf{w}_{fp_2}^T \mathbf{x}_{fp_1}$, both scores are calculated and included in the experiments separately as either genuine or imposter matching scores.

Equal error rates (EERs) for GMMs with different number of Gaussians are shown in Table 6.2. The optimal C values for training SVMs corresponding to different number of Gaussians are found by grid search and best C values were 10, 0.001, and 1 for 1024, 2048, and 4096 Gaussians, respectively. As the number of Gaussians in the GMMs increases, our method performs better in representing feature distribution which eventually leads to lower error rates.

In order to provide comparison with our system, we also perform direct minutiae matching[4] with the commercial algorithm which we also use for minutiae extraction. It obtains 0.50% EER on FVC2002DB1A and performs better than our method. This difference stems from the facts that we can neither perform minutiae pair search, which is a crucial step for minutiae matching, nor include singular point information. However, the main purpose of this study is to present a fixed-length fingerprint representation and this performance drop is expected.

6.3.2 Discussion

The GMM-SVM based feature representation is a novel method to create a fixed-length feature vector for fingerprint minutiae. Although minutiae-based matching is the most widely used technique in fingerprint verification/identification, the increasing security and privacy concerns make minutiae template protection one of the most crucial tasks. The main motivation of this study is to obtain a fixed-length feature vector for fingerprints so that minutiae based fingerprint verification can be combined with template protection schemes. In addition, our method avoids the difficulties of minutiae registration by representing minutiae patches on a normalized coordinate system defined by the

[4] Additional fingerprint features that are not defined in ISO minutiae template are not used in any of the experiments.

orientation of the central minutia. Also, major problem of elastic distortion in fingerprint matching is compensated with the local representation of the minutiae neighborhoods.

This study introduces a fixed-length feature representation for variable length minutiae of a fingerprint. In order to combine our method with the cryptographic primitives for template protection, such as [71, 72], one should extract bits that are stable for genuine users and completely random for arbitrary users. Random projection-based biometric hashing [7] cannot be directly applied to minutiae templates, however, another possible direction for securing minutiae might be applying biohashing to our GMM-SVM fingerprint feature vectors.

In the remaining of this chapter, we will work towards hashing of the feature vectors created by our approach and include the binarization of the GMM-SVM feature vectors. We also conjecture that enriching the database that is used in training GMMs and using random resampling ([73]) for addressing data-imbalance problem in SVM will be possible improvements to our GMM-SVM minutiae representation.

6.4 Asymmetric Locality Sensitive Hashing

In this section, we introduce an asymmetric hashing scheme for the inner product matching of GMM-SVM feature vectors presented in Section 6.3. The matching score between a reference SVM model template and a query GMM supervector is calculated via inner product of these two vectors. The main goal of asymmetric locality sensitive hashing is to convert reference SVM model templates and query GMM supervectors into binary strings, so that they can be compared using Hamming difference. The asymmetry comes from the differences in transformations that are applied on template model vectors and query fingerprint feature vectors separately. We adopt the asymmetric locality sensitive hashing method proposed in [19] for maximum inner product search using sign random projections. First, we present the fundamentals of locality sensitive hashing. Then, we continue with the asymmetric feature transformation.

6.4.1 Locality Sensitive Hashing

Locality sensitive hashing (LSH) [74] is initially proposed to solve the problem of efficiently finding nearest neighbors. It improves over the brute-force algorithm in which the query point is compared to each data point. LSH is a family of functions with the property that more similar items in the d-dimensional Euclidean space according to some similarity measure have a higher collision probability hence a lower expected Hamming distance (Figure 6.6).

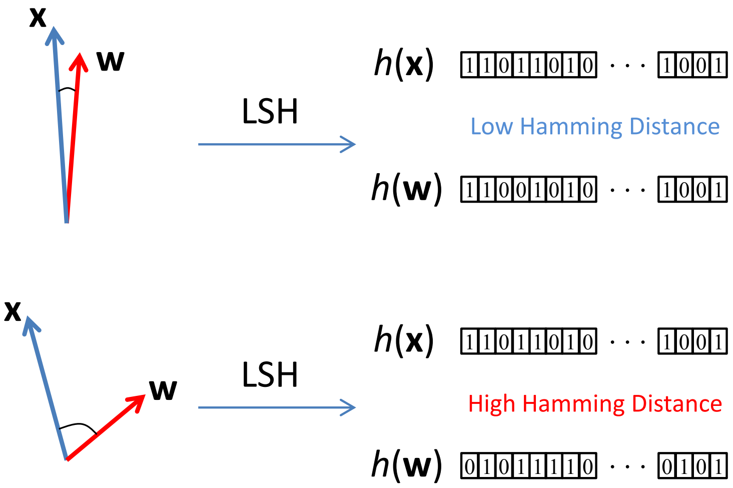

x
w
LSH
h(x) 1 1 0 1 1 0 1 0 · · · 1 0 0 1
Low Hamming Distance
h(w) 1 1 0 0 1 0 1 0 · · · 1 0 0 1
x
w
LSH
h(x) 1 1 0 1 1 0 1 0 · · · 1 0 0 1
High Hamming Distance
h(w) 0 1 0 1 1 1 1 0 · · · 0 1 0 1

FIGURE 6.6: Illustration of the LSH scheme

6.4.1.1 LSH for correlation

In our GMM-SVM framework,the inner product between a query vector $\mathbf{x}$ and a database model $\mathbf{w}$, $\mathbf{w}^{\mathbf{T}}\mathbf{x}$ is the score used for making decisions. It is required to find a hashing scheme such that the Hamming distance between the hashes of the vectors would approximate their inner product.

Sign random projection (SRP) is a popular LSH family in which the sign of the projection is kept [76, 77]. The hash function using SRP is defined as:

$$h_t(\mathbf{x}) = \mathrm{sign}(\mathbf{a}_t^T \mathbf{x}), \tag{6.9}$$

where $\mathbf{a}_t$ is randomly chosen from i.i.d. normal distribution with $\mathbf{a}_t(i) \sim N(0,1)$. Note that, this is the same as biohashing. So, we are using biohashing for LSH here.

The correlation or cosine similarity between a query $\mathbf{x}$ and model $\mathbf{w}$ is defined as:

$$\frac{\mathbf{w}^T\mathbf{x}}{\|\mathbf{w}\|\|\mathbf{x}\|}. \tag{6.10}$$

It has been shown that the correlation between vectors is monotonically related to the collision probability of sign random projections [77] due to the following relation:

$$P(h_t(\mathbf{x}) = h_t(\mathbf{w})) = 1 - \frac{1}{\pi}cos^{-1}\left(\frac{\mathbf{w}^T\mathbf{x}}{(||\mathbf{w}||||\mathbf{x}||)}\right). \tag{6.11}$$

So, the probability of one bit random hashes of two vectors to be equal is monotonically related to the cosine similarity (or correlation) between two vectors. After a basic transformation of the hash code outputs from $\{-1,1\}$ to $\{0,1\}$, the Hamming distance between two N-bit hashes in terms of XOR is:

$$D_h(h(\mathbf{w}), h(\mathbf{x})) = \sum_{t=1}^{N}\left(h_t(\mathbf{w}) \oplus h_t(\mathbf{x})\right), \tag{6.12}$$

where D_h denotes the Hamming distance operator and $\oplus$ is the bitwise XOR operator. Due to this relation, Hamming distance can be seen as a sum of Bernoulli random variables and consequently has a binomial distribution with probability of success equal to one minus the collision probability given in 6.11. Here, we make a simplifying assumption of independent bits.

Hence for an N-bit SRP hash, the expected value of the Hamming distance between the hashes is:

$$E(D_h(h(\mathbf{x}), h(\mathbf{w}))) = \frac{N}{\pi}cos^{-1}\left(\frac{\mathbf{w}^T\mathbf{x}}{(||\mathbf{w}||||\mathbf{x}||)}\right). \tag{6.13}$$

The relation between vector correlations which takes values in the range [-1,1] and the expected N-bit normalized Hamming distance between their SRP hashes is shown in Figure 6.7. The confidence intervals shown in this figure is obtained by assuming that the Hamming distance is distributed binomial. It is easy to see that if the number of bits N

is large enough, correlation and N-bit Hamming distances will be monotonically related, hence an ascending ranking of expected Hamming distances and a descending ranking of correlations will be the same. This means that if we wanted to use correlations between vectors for scoring verification attempts, we can safely use Hamming distances between N-bit SRP hashes instead, without significantly affecting verification performance.

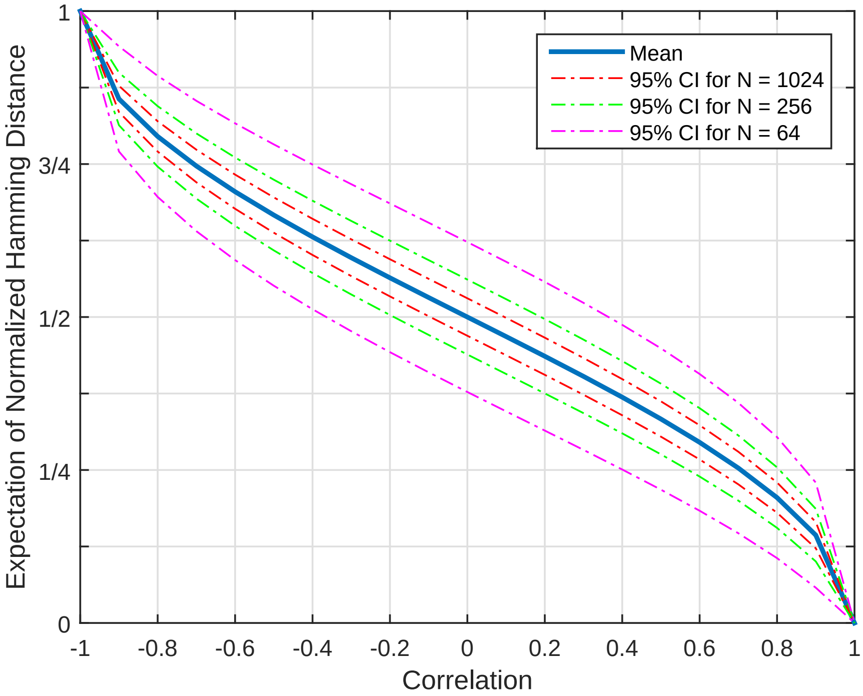


FIGURE 6.7: Relation between the correlation and the expected normalized Hamming distance together with 95% confidence intervals for different hash lengths

However, we are interested in calculating inner products and not correlations between vectors due to our linear SVM formulation. In order to be able to approximate inner products with Hamming distances, we need to use asymmetric hashing.

6.4.2 Asymmetric Feature Transformation

In our GMM-SVM framework, matching score between a query fingerprint and a reference template is calculated based on the inner product of the corresponding vectors. Due to variations in the norms of these vectors, an inner product cannot be substituted with a correlation directly. Asymmetric feature transformation, where the transformations $F_{\mathbf{x}}$ and $F_{\mathbf{w}}$ are different for input query and reference data, converts an inner product to a correlation so that outputs of these two operations are approximately equal [19].

It should be noted that the preprocessing transformation $F_{\mathbf{x}}$ is applied on the query $\mathbf{x}$ and the preprocessing transformation $F_{\mathbf{w}}$ is applied on the reference model $\mathbf{w}$.

For the purpose of converting correlations to inner products, [19] defines two vector transformations $F'_{\mathbf{w}} : \mathbb{R}^D \mapsto \mathbb{R}^{D+2m}$ and $F'_{\mathbf{x}} : \mathbb{R}^D \mapsto \mathbb{R}^{D+2m}$ as:

$$F'_{\mathbf{w}}(\mathbf{w}) = [\mathbf{w}; 1/2 - \|\mathbf{w}\|_2^2; 1/2 - \|\mathbf{w}\|_2^4; ...; 1/2 - \|\mathbf{w}\|_2^{2m}; 0; 0; ...; 0], \tag{6.14}$$

$$F'_{\mathbf{x}}(\mathbf{x}) = [\mathbf{x}; 0; 0; ...; 0; 1/2 - \|\mathbf{x}\|_2^2; 1/2 - \|\mathbf{x}\|_2^4; ...; 1/2 - \|\mathbf{x}\|_2^{2m}]. \tag{6.15}$$

The transformation $F'_{\mathbf{w}}(\mathbf{w})$ first appends m components of the form $1/2 - \|\mathbf{w}\|_2^{2^i}$ and then m zeros. Its asymmetric counterpart $F'_{\mathbf{x}}(\mathbf{x})$ first appends m zeros and then m components of the form $1/2 - \|\mathbf{x}\|_2^{2^i}$.

Without loss of generality, it can be assumed that $\|\mathbf{w}_i\|_2 \leq U < 1, \forall \mathbf{w}_i$ enrolled in the database and $\|\mathbf{x}_i\|_2 \leq U < 1, \forall \mathbf{x}_i$. If that is not the case, it is possible to scale all data points. Let M be the upper bound on all norms, i.e. $M = \max(\max \|\mathbf{w}\|_2, \max \|\mathbf{x}\|_2)$ and the transformation $T : \mathbb{R}^D \mapsto \mathbb{R}^D$ as:

$$T(\mathbf{x}) = \mathbf{x}\frac{U}{M}. \tag{6.16}$$

We apply this transformation first to limit the norms of the vectors, then we apply the asymmetric transforms provided in 6.14 and 6.15. Finally, we obtain combined transformations $F_{\mathbf{w}}$ and $F_{\mathbf{x}}$ which can be defined as $F_{\mathbf{w}} = F'_{\mathbf{w}} \circ T$ and $F_{\mathbf{x}} = F'_{\mathbf{x}} \circ T$, respectively. The inner product between a query $\mathbf{x}$ and reference data $\mathbf{w}$ after the transformations $F_{\mathbf{w}}$ and $F_{\mathbf{x}}$ become:

$$F_{\mathbf{w}}(\mathbf{w})^T F_{\mathbf{x}}(\mathbf{x}) = \mathbf{w}^T \mathbf{x} \left(\frac{U^2}{M^2}\right). \tag{6.17}$$

$F_{\mathbf{w}}(\mathbf{w})$ and $F_{\mathbf{x}}(\mathbf{x})$ are now $D + 2m$ dimensional and their norms are given by:

$$\|F_{\mathbf{w}}(\mathbf{w})\| = \sqrt{\frac{m}{4} + \|T(\mathbf{w})\|_2^{2m+1}}, \tag{6.18}$$

$$\|F_{\mathbf{x}}(\mathbf{x})\| = \sqrt{\frac{m}{4} + \|T(\mathbf{x})\|_2^{2m+1}}. \tag{6.19}$$

After the asymmetric transformations on $\mathbf{w}$ and $\mathbf{x}$, the correlation between $F_{\mathbf{w}}(\mathbf{w})$ and $F_{\mathbf{x}}(\mathbf{x})$ is:

$$Corr\left(F_{\mathbf{w}}(\mathbf{w}), F_{\mathbf{x}}(\mathbf{x})\right) = \frac{\mathbf{w}^T\mathbf{x}\left(\frac{U^2}{M^2}\right)}{\sqrt{\frac{m}{4} + \|T(\mathbf{w})\|_2^{2m+1}}\sqrt{\frac{m}{4} + \|T(\mathbf{x})\|_2^{2m+1}}}. \tag{6.20}$$

The norms $\|T(\mathbf{w})\|_2$ and $\|T(\mathbf{x})\|_2$ are less than 1. Therefore, both $\|T(\mathbf{w})\|_2^{2m+1}$ and $\|T(\mathbf{x})\|_2^{2m+1}$ converge to zero very fast when m is chosen to be large enough and the correlation approximately becomes proportional to the inner product $\mathbf{w}^T\mathbf{x}$. The sign random projection can then be applied on $F_{\mathbf{w}}(\mathbf{w})$ and $F_{\mathbf{x}}(\mathbf{x})$ to generate respective hashes. The Hamming distance between the asymmetric hashes of $\mathbf{w}$ and $\mathbf{x}$ is calculated as:

$$\sum_{t=1}^{N}\left(h_t(F_{\mathbf{w}}(\mathbf{w})) \oplus h_t(F_{\mathbf{x}}(\mathbf{x}))\right). \tag{6.21}$$

This Hamming distance is monotonically related to the correlation between transformed vectors due to Equation 6.13 when number of bits N is sufficiently large, which in turn is proportional to the inner product of $\mathbf{w}^T\mathbf{x}$ as given in Equation 6.20 when m is sufficiently large. This means that we can use Hamming distance between these asymmetric hashes instead of using inner products for making verification decisions.

6.4.3 Experiments and Results

We perform the same one-to-one fingerprint verification experiments on the FVC2002DB1A database as in Section 6.3.1. Using the GMM supervectors as query fingerprint vectors and SVM model templates as reference data, we conduct asymmetric hashing experiments by following the verification protocol described in Section 6.3.1 where report EERs for matching of GMM-SVM vectors are reported with 1024, 2048, and 4096 number of Gaussians as 2.231%, 1.911%, and 1.633%, respectively. Table 6.3 presents the verification EERs for asymmetric hashing.

# Gaussians	GMM-SVM	ALSH
1024	2.191%	2.362%
2048	1.911%	1.850%
4096	1.633%	1.661%

Table 6.3: Equal error rates for Asymmetric Locality Sensitive Hashing for correlation

The results show that the verification performance is not altered by the asymmetric transformation followed by hashing. The error rates are very close to each other for matching of the GMM-SVM vectors and ALSH hash vectors.

6.5 Improvements

In this section, we present two modifications that improve efficiency and accuracy of the proposed fingerprint matching approach. First, we reduce the dimension of the GMM-SVM feature vectors via Principal Component Analysis (PCA) [46]. Next, we use random minutiae resampling for addressing data-imbalance problem in SVM to improve our GMM-SVM minutiae representation.

6.5.1 Dimension Reduction with PCA before SVM Training

The GMM-SVM features are DN dimensional, where $D = 50$ is the number of DCT coefficients and $N = 1024, 2048$, or 4096 is the number of Gaussians in the mixture. Respectively, this corresponds to feature lengths of 51200, 102400, and 204800 which are quite high for computational complexity of the SVM training process and matching step. Thus, we apply PCA on the input features and reduce the dimension of $D \times N$ to 799. The selection of 799 is due to the nature of the available data. FVC2002DB1 database includes 800 fingerprint samples (8 impression of 100 different fingers) and the highest number of corresponding principal components is 800-1=799.

6.5.2 Random Minutiae Sampling for SVM Training

There is an evident imbalance between number of positive and negative samples for SVM training at the reference template generation step. For every enrollment sample, there is only 1 positive sample which corresponds to the given enrollment fingerprint

GMM supervector. On the other hand, we have 99 negative samples that come from 99 different fingerprints available in the database. So, the training data for SVM has 100 samples and only 1 of them has a positive label (i.e. +1) and the remaining 99 samples are labeled as negative (i.e. −1). Due to this data imbalance, the orientation of the decision boundary is largely dictated by the negative samples in the training data.

In order to address the problem of data imbalance, we follow the work in [73] which proposes to increase the number of positive samples by randomly selecting a subset of the input features. Specifically, the order of minutiae in an enrollment fingerprint is first randomized; then random subsets of minutiae are selected among this random set. Each of these subsets is then used to produce a GMM supervector. A desirable number of fingerprint supervectors can be produced by repeating this randomization and partitioning process a number of times.

Given an enrollment fingerprint, we randomize the order of minutiae and selected the first $n\%$ minutiae to create a subset. This procedure is repeated for different selections of percentage n, and several subsets of minutiae are generated. Then, GMM supervectors corresponding to each subset is created and included in the SVM training as positive training samples, in addition to the GMM supervector of the original enrollment fingerprint full length minutiae set.

Table 6.4 shows the number of random sampling repetitions for different values of n. A total of 16 subsets are generated and when combined with the original full length minutiae set, 17 positive samples become available for SVM training from GMM supervectors.

n%	**# repetitions**
95%	1
90%	2
85%	3
80%	4
75%	3
70%	2
65%	1
TOTAL	**16**

Table 6.4: Number of subsets selected for different percentages of minutiae

The implemented improvements significantly decreases error rates of the GMM-SVM approach (Table 6.5). Equal error rates usually decrease when the number of hash dimension is higher. However, increasing the hash dimension even more does not lead to lower error rates and usually hash dimensions between $2^{18} - 2^{22}$ are enough to obtain the best accuracies (Figure 6.8). For brevity, the table do not include EERs for every hash dimension. Instead, we report the results for hash dimension of 2^{26}. In addition, FAR1000 values (i.e., FRR values where FAR $= 10^{-3}$) are given in Table 6.6.

# Gaussians	**GMM-SVM initial**	**GMM-SVM improved**	**Asymmetric Transformation**	**ALSH - 2^{26}**
1024	2.191%	1.180%	1.118%	1.123%
2048	1.911%	1.175%	1.024%	1.010%
4096	1.633%	1.218%	1.175%	1.180%

TABLE 6.5: Equal error rates of the improved system for FVC2002DB1A

# Gaussians	**ALSH - 2^{26}**	
	EER	**FAR1000**
1024	1.123%	2.625%
2048	1.010%	2.500%
4096	1.180%	2.696%

TABLE 6.6: FAR1000 values of the improved ALSH scheme for FVC2002DB1A

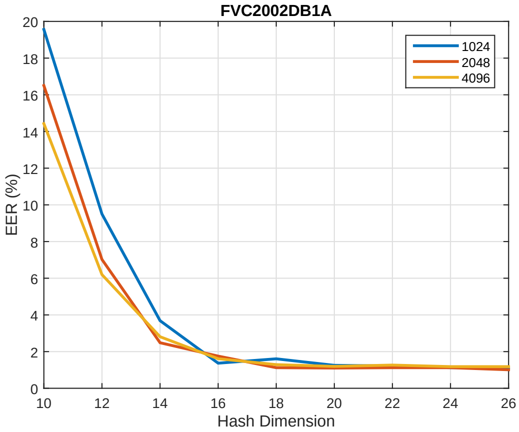
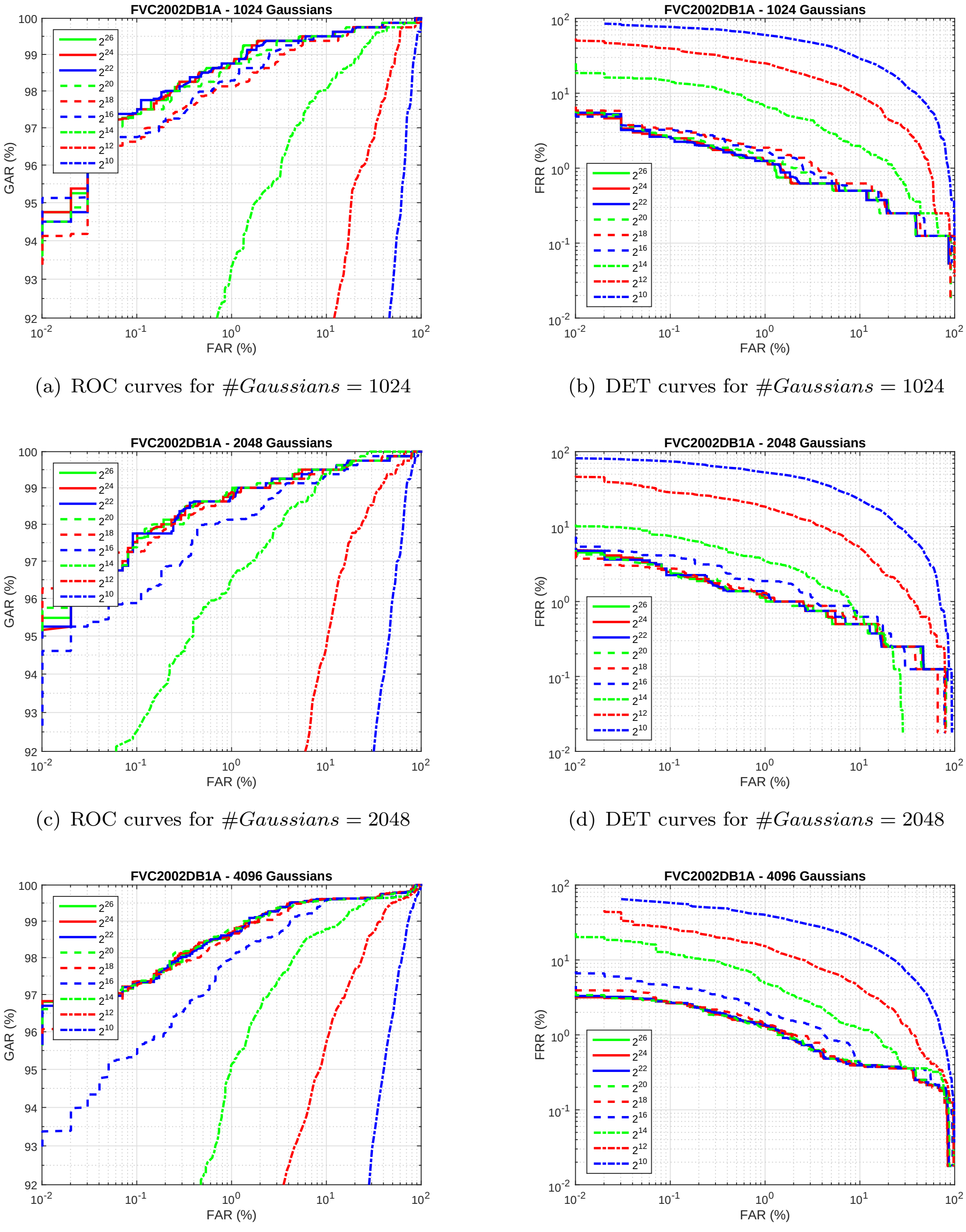


FIGURE 6.8: EERs for different hash dimensions - FVC2002DB1A

Figures 6.9(a) to 6.9(f) illustrate the corresponding ROC and DET curves for systems with different number of Gaussians. For every implementation of the system, i.e., with



(a) ROC curves for $\#Gaussians = 1024$

(b) DET curves for $\#Gaussians = 1024$

(c) ROC curves for $\#Gaussians = 2048$

(d) DET curves for $\#Gaussians = 2048$

(e) ROC curves for $\#Gaussians = 4096$

(f) DET curves for $\#Gaussians = 4096$

FIGURE 6.9: Error rates for FVC2002DB1A

different number of Gaussians in the mixture, reducing the feature dimension and increasing the number of positive samples for SVM training improves the robustness of the system and decreases matching errors.

Figure 6.10 illustrates the hash generation time for the ALSH scheme, i.e., the time for generation of a bit-string from a query or reference GMM-SVM feature vector. The

GMM-SVM feature generation and ALSH scheme are implemented in MATLAB and the experimens are run on a 2.5 GHz with 64 GB of RAM PC using 64-bit Windows Server 2008 operating system. It should be noted that the final output of the system is a bit-string, therefore the matching step is low cost and suitable for light weight applications such as match-on-card systems. Figure 6.11 illustrates the matching time between two hashes for different hash lengths. Even if the hash length is 2^{26}, the matching time is less than 0.2 seconds, which is acceptable for a fingerprint verification system.

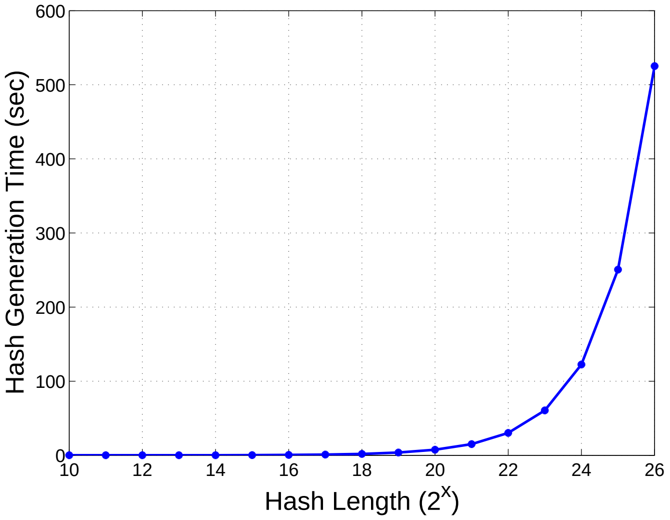


FIGURE 6.10: ALSH hash generation time from GMM-SVM feature

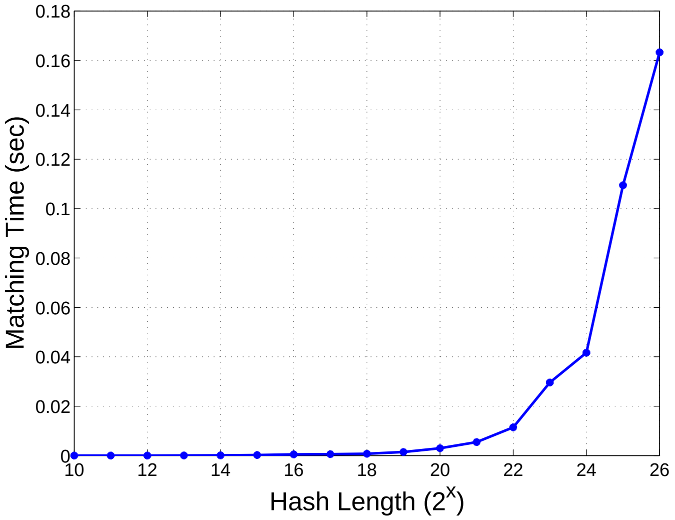


FIGURE 6.11: ALSH hash matching time

The verification performance of our approach is also validated on another fingerprint database, namely FVC2002DB2A. Table 6.7 includes the EERs corresponding to different number of Gaussians and presents error rates for different steps of the framework at each column. In addition, FAR1000 values (i.e., FRR values where FAR = 10^{-3}) are given in Table 6.8. Figure 6.12 presents the EERs for different hash dimensions and the corresponding ROC and DET curves are given in Figures 6.13(a) to 6.13(f).

# Gaussians	**GMM-SVM**	**Asymmetric Transformation**	**ALSH - 2^{22}**
1024	1.834%	1.645%	1.664%
2048	1.426%	1.374%	1.393%
4096	1.123%	1.118%	1.000%

Table 6.7: Equal error rates of the improved system for FVC2002DB2A

# Gaussians	**ALSH - 2^{22}**	
	EER	**FAR1000**
1024	1.664%	4.286%
2048	1.393%	3.018%
4096	1.000%	2.750%

Table 6.8: FAR1000 values of the improved ALSH scheme for FVC2002DB2A

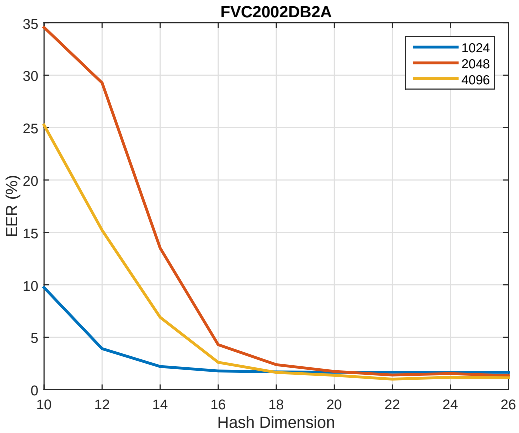
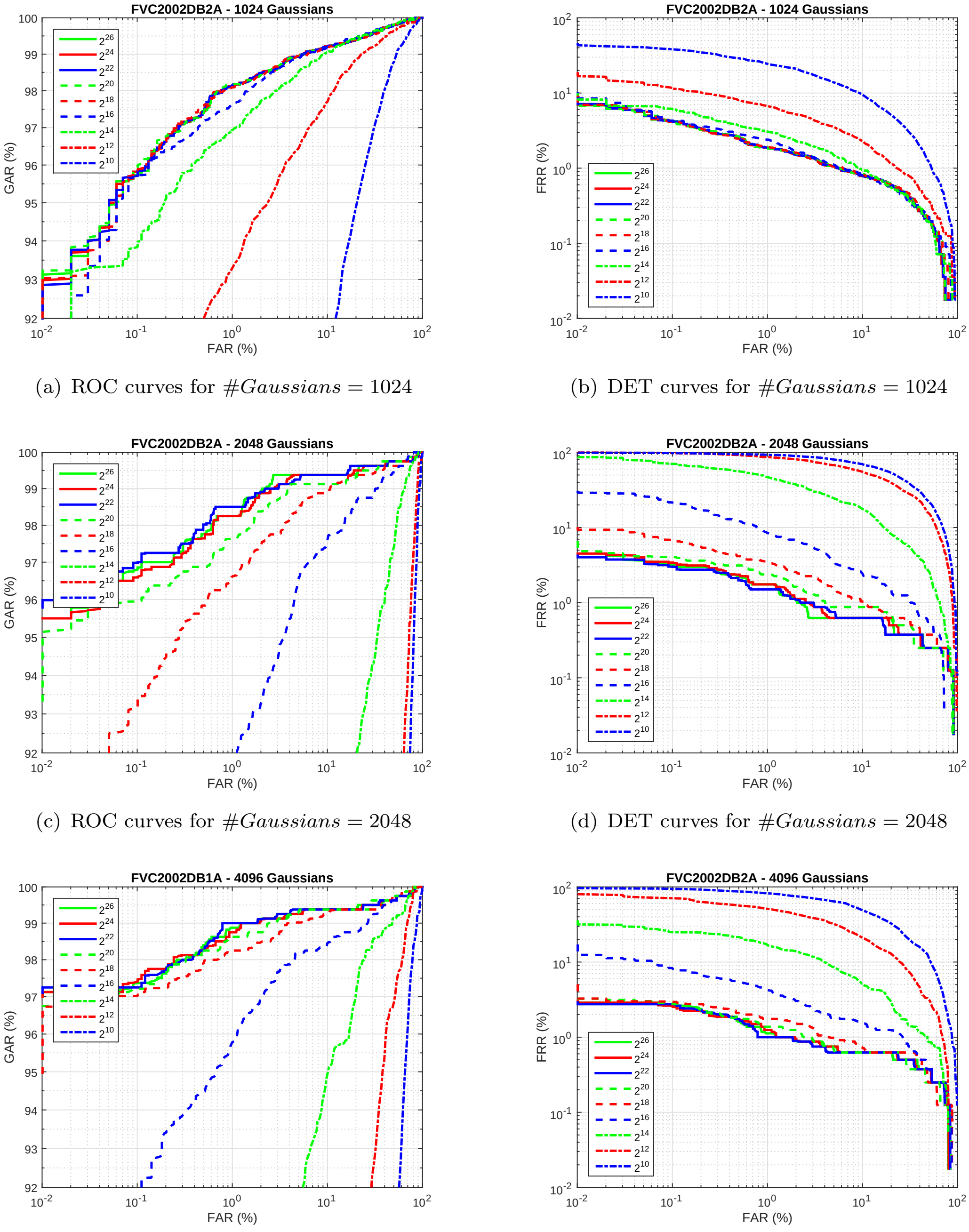


Figure 6.12: EERs for different hash dimensions - FVC2002DB2A

Two other fixed length approaches that can be compared with our system are the spectral minutiae representation [17] and binary feature vector representation in [38]. However, they do not report error rates for the FVC2002DB1A database. When we analyze their reported results on the FVC2002DB2A database (2.48% [17] and 3.88% [38] EERs compared to minutiae matching 1.0% on FVC2002DB2A), we also observe similar performance drops compared to minutiae matching. The best performance obtained for



(a) ROC curves for $\#Gaussians = 1024$

(b) DET curves for $\#Gaussians = 1024$

(c) ROC curves for $\#Gaussians = 2048$

(d) DET curves for $\#Gaussians = 2048$

(e) ROC curves for $\#Gaussians = 4096$

(f) DET curves for $\#Gaussians = 4096$

FIGURE 6.13: Error rates for FVC2002DB2A

FVC2002DB2A with our approach is 1.000%, which is far lower than those two results. It is also at the same level with minutiae matching result on the same database.

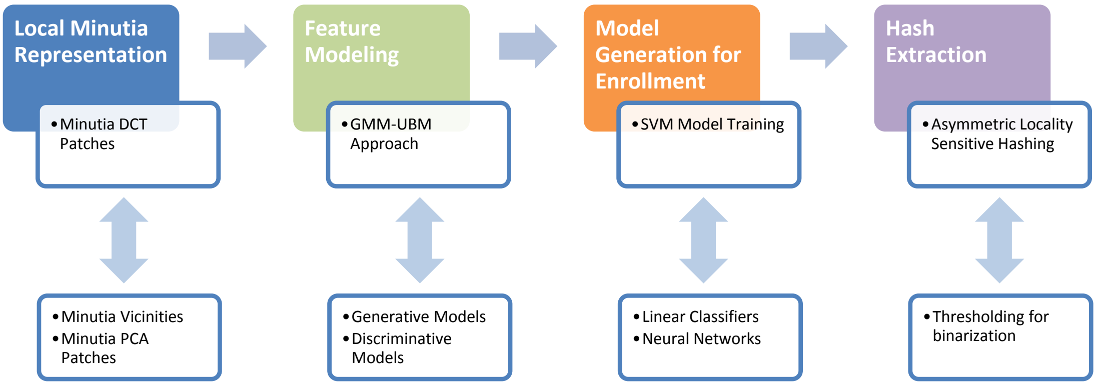


Figure 6.14: Overall framework and other possibilities

6.6 Flexibility of the Framework - Enabling Other Possibilities

The overall framework consists of four separate components and each component has its own objective. This allows various other possible methods to either be integrated to the system or replace the implemented ones (Figure 6.14).

The first component aims at representing local minutiae information in a compact way. For this task, we propose using DCT coefficients of the minutia patch images. It is also possible to use other image descriptors such as PCA or LBP (Local Binary Patterns). Also, one does not have to use a minutia patch image at all. Instead, local minutiae constructs such as minutia vicinities, minutiae triplets or MCC (Minutiae Cylinder Codes) features can be used at this stage.

Feature modeling is the second component and it aims at estimating the distribution of the input local minutia vectors. In our framework, this has been accomplished by GMM-UBM approach and each fingerprint is represented according to its relative alignment into the background Gaussian mixture. In order to model the input feature distribution, other alternatives among parametric/non-parametric generative or discriminative models can also be employed.

The next component generates a model template for a given enrollment fingerprint sample. At this stage, we train a linear SVM for each enrollment sample. The main aim at this step is to create a model for an enrollment sample that would discriminate samples of the same fingerprint from other imposter fingerprint samples. A collection of linear classifiers are available at this stage and can be used in the same context as well. In addition, Neural Networks can be another direction to discriminate positive samples from negative ones.

The last step includes the binarization of the resulting fixed-length feature vectors. Binary representation of the features not only leads to faster matching, but also allows combination of the fingerprint templates with cryptographic primitives based on homomorphic encryption and opens an alternative path to biometric template protection. We propose to use asymmetric locality sensitive hashing at this stage and extract binary strings, or namely hashes, that can easily be compared via Hamming distance. Other possible thresholding strategies can also be followed here.

6.7 Discussion

In this chapter, we present a novel framework for fixed-length feature generation from fingerprint minutiae. As each fingerprint has varying number of minutiae, it is a critical bottleneck for fingerprint template protection to obtain a fixed-length representation. Most of the current fingerprint verification systems use only minutiae information since minutiae representation is globally regarded as the standard feature for fingerprint matching. Therefore, other additional features that can be extracted from fingerprints, such as ridge information, orientation, texture, etc., are not included in this study. Singular points (core and delta) are also included in the standard minutiae formats (i.e. ISO 19794-2 [60]), however their automatic detection can be misleading. Also, not all fingerprint images include singular points due to exaggerated displacement. So, we keep singular points out of discussion in this work.

In order to address the security and privacy concerns regarding protection of the minutiae templates, we propose a multi step feature generation framework based on GMM-SVM approach. The last step of the framework includes the binarization of the obtained fingerprint features without decreasing the representative ability while keeping the representation as compact as possible. Asymmetric locality sensitive hashing provides two separate transformations. One of them is applied on query fingerprint templates or the other one on reference data that is previously stored in the system. These transformations allow the use of locality sensitive hashing for converting the GMM-SVM feature vector into binary hash vectors. The experiments conducted on publicly available fingerprint databases show the success of our system and its superiority over existing fixed-length minutiae representations.

Chapter 7

Conclusion and Future Work

Biometric template protection methods are natural extensions of existing biometric recognition systems. These methods aim at securing the templates of system users that are either saved in smart cards or large biometric databases depending on the system design. The security and privacy of biometric templates are of greatest importance and attacks to biometric systems and databases severely threaten the security and privacy of the society.

This dissertation evaluates one of the current biometric template protections methods, namely biometric hashing, from security and privacy aspects. Thorough analysis of biometric hashing requires theoretical evaluation of the method as well as analysis of practical attacks. In this study, we theoretically analyze the unpredictability of biohashes via estimated entropy and the amount of information carried in a biohash is measured for the first time. In addition, several inversion attacks are proposed and weaknesses of biometric hashing is discussed. Thus, a complete assessment of biometric template protection with biometric hashing is presented.

Fingerprint modality stands out from other biometric modalities due its distinct properties. The standard matching of fingerprints depends on minutiae features which have a non-constant length by nature. This does not allow current biometric template protection methods to be applied for securing minutiae templates. Spectral minutiae representation is one of the previous methods proposed to provide a fixed-length feature vector for fingerprint minutiae. This work includes an evaluation of this method and a potential template protection for spectral minutiae via biometric hashing.

This dissertation presents a novel framework to bypass the bottleneck of varying length input feature to template protection by generating a fixed-length representation of fingerprint minutiae. This task requires two subtasks to be completed. First of all, discriminative features that represent the local minutia structures has to be extracted. Second, the distribution of such local features should be modeled. Local minutia information is captured by a minutia patch image and expressed via the first D DCT coefficients of this image. The GMM-SVM approach, which was first applied to speaker verification, is adapted to fingerprint verification based on minutiae features with some modifications. Given a fingerprint sample, its minutiae information is converted into a fixed-length feature vector and become securable with a biometric template protection method such as fuzzy commitment and biometric hashing. Furthermore, the binarization of feature vectors via asymmetric locality sensitive hashing enables the combination of homomorphic encryption based cryptographic alternatives and minutiae information.

7.1 Evaluation of Biometric Hashing

Biometric hashing provides an intelligent solution for protecting biometric templates and thus deserves considerable attention. However, vanilla biohashing has significant security and privacy drawbacks, as discussed in Chapters 3 and 4. Especially, if the secret key of a user is known to an adversary, there is a significant drop in the entropy of biohashes. So, novel random projection and quantization schemes are required to prevent or limit this drop. However, entropy is not the only theoretical metric in analyzing biometric template protection methods. Other privacy and security metrics should also be investigated and ways to implement them on biometric hashing should be studied.

The reconstruction of the biometric template or the original signal from biohash is another security and privacy aspect of biometric hashing as discussed in Chapter 4. Therefore, better biohashing schemes that address the improvements in theoretical security should also be studied in order to provide resistance against inversion attacks.

7.2 Template Protection for Fingerprint Minutiae

One of the main supporting ideas of this dissertation is that a fixed-length representation for minutiae is required for fingerprint template protection. Chapters 5 and 6 discuss this requirement in depth and propose novel solutions. The GMM-SVM framework is able to fill this gap by offering a fixed-length representation which can be combined with existing template protection methods such as fuzzy commitment and biohashing. A possible future work would be investigating applicable template protection methods and securing fingerprint minutiae templates using GMM-SVM features.

Asymmetric locality sensitive hashing provides a successful binary representation for GMM-SVM features. This allows the use of cryptographic primitives and homomorphic encryption for securing minutiae information. A promising area of future work would be building novel encryption strategies for securing binary minutiae features of this framework.

Bibliography

[1] John Daugman, "The importance of being random: statistical principles of iris recognition," *Pattern Recognition*, vol. 36, no. 2, pp. 279–291, 2003.

[2] Davide Maltoni, Dario Maio, Anil K. Jain, and Salil Prabhakar, *Handbook of Fingerprint Recognition*, Springer Publishing Company, Incorporated, London, 2nd edition, 2009.

[3] Tanya Ignatenko and Frans M. J. Willems, "Information leakage in fuzzy commitment schemes," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 337–348, 2010.

[4] Salil Prabhakar, Sharath Pankanti, and Anil K. Jain, "Biometric recognition: Security and privacy concerns," *IEEE Security & Privacy*, vol. 1, no. 2, pp. 33–42, 2003.

[5] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle, "An analysis of minutiae matching strength," in *Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication*, London, UK, 2001, pp. 223–228, Springer-Verlag.

[6] Ari Juels and Martin Wattenberg, "A fuzzy commitment scheme," in *Proceedings of the 6th ACM Conference on Computer and Communications Security*, New York, NY, USA, 1999, CCS '99, pp. 28–36, ACM.

[7] Andrew Teoh Beng Jin, David Ngo Chek Ling, and Alwyn Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognition*, vol. 37, no. 11, pp. 2245 – 2255, 2004.

[8] Xuebing Zhou, "Privacy and security assessment of biometric template protection," *IT - Information Technology*, vol. 54, no. 4, pp. 197–200, 2012.

[9] David Ngo Chek Ling, Andrew Teoh Beng Jin, and Alwyn Goh, "Biometric hash: high-confidence face recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 6, pp. 771–775, June 2006.

[10] Tee Connie, Andrew Teoh Beng Jin, Michael Goh Kah Ong, and David Ngo Chek Ling, "Palmhashing: a novel approach for cancelable biometrics," *Information Processing Letters*, vol. 93, no. 1, pp. 1–5, 2005.

[11] Abhishek Nagar, Karthik Nandakumar, and Anil K. Jain, "Biometric template transformation: A security analysis," in *Media Forensics and Security*. 2010, vol. 7541 of *SPIE Proceedings*, p. 75410, SPIE.

[12] Bian Yang, Christoph Busch, Patrick Bours, and Davrondzhon Gafurov, "Robust minutiae hash for fingerprint template protection," in *Media Forensics and Security*. 2010, vol. 7541 of *SPIE Proceedings*, p. 75410, SPIE.

[13] Karl Kummel, Claus Vielhauer, Tobias Scheidat, Dirk Franke, and Jana Dittmann, "Handwriting biometric hash attack: A genetic algorithm with user interaction for raw data reconstruction," in *Communications and Multimedia Security*. 2010, vol. 6109 of *Lecture Notes in Computer Science*, pp. 178–190, Springer.

[14] Yi C. Feng, Meng-Hui Lim, and Pong C. Yuen, "Masquerade attack on transform-based binary-template protection based on perceptron learning," *Pattern Recognition*, vol. 47, no. 9, pp. 3019–3033, 2014.

[15] Anil K. Jain, Karthik Nandakumar, and Abhishek Nagar, "Biometric template security," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, pp. 113:1–113:17, Jan. 2008.

[16] Haiyun Xu, Raymond N.J. Veldhuis, Asker M. Bazen, Tom A.M. Kevenaar, Ton A.H.M. Akkermans, and Berk Gökberk, "Fingerprint verification using spectral minutiae representations," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 3, pp. 397–409, 2009.

[17] Haiyun Xu and Raymond N. J. Veldhuis, "Complex spectral minutiae representation for fingerprint recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2010, San Francisco, CA, USA, 13-18 June, 2010*. 2010, pp. 1–8, IEEE.

[18] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, pp. 308–311, 2006.

[19] Anshumali Shrivastava and Ping Li, "Improved asymmetric locality sensitive hashing (ALSH) for maximum inner product search (MIPS)," in *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI 2015, July 12-16, 2015, Amsterdam, The Netherlands*, 2015, pp. 812–821.

[20] Umut Uludag, Sharath Pankanti, and Anil K. Jain, "Fuzzy vault for fingerprints," in *Audio- and Video-Based Biometric Person Authentication*, Takeo Kanade, Anil K. Jain, and Nalini K. Ratha, Eds., vol. 3546 of *Lecture Notes in Computer Science*, pp. 310–319. Springer Berlin Heidelberg, New York, NY, USA, 2005.

[21] Bian Yang, Daniel Hartung, Koen Simoens, and Christoph Busch, "Dynamic random projection for biometric template protection," in *Fourth IEEE International Conference on Biometrics: Theory Applications and Systems, BTAS 2010, Washington, DC, USA, 27-29 September, 2010*, 2010, pp. 1–7.

[22] Adams Kong, King Hong Cheung, David Zhang, Mohamed S. Kamel, and Jane You, "An analysis of biohashing and its variants," *Pattern Recognition*, vol. 39, no. 7, pp. 1359–1368, 2006.

[23] Xuebing Zhou and Ton Kalker, "On the security of biohashing," in *Media Forensics and Security*, United States, 2010, vol. 7541 of *SPIE Proceedings*, p. 75410, SPIE.

[24] Oded Goldreich, *Foundations of Cryptography: Volume 1*, Cambridge University Press, New York, NY, USA, 2006.

[25] King Hong Cheung, Adams Wai-Kin Kong, Jane You, and David Zhang, "An analysis on invertibility of cancelable biometrics based on biohashing," in *Proceedings of The 2005 International Conference on Imaging Science, Systems, and Technology: Computer Graphics, CISST*, Hamid R. Arabnia, Ed., Las Vegas, Nevada, USA, 2005, pp. 40–45, CSREA Press.

[26] Yongjin Lee, Yunsu Chung, and Kiyoung Moon, "Inverse operation and preimage attack on biohashing," in *IEEE Workshop on Computational Intelligence in Biometrics: Theory, Algorithms, and Applications, CIB*, March 2009, pp. 92–97.

[27] Patrick Lacharme, Estelle Cherrier, and Christophe Rosenberger, "Preimage attack on biohashing," in *International Conference on Security and Cryptography, SECRYPT*, Pierangela Samarati, Ed., Reykjavk, Iceland, 2013, pp. 363–370, SciTePress.

[28] Ari Juels and Madhu Sudan, "A fuzzy vault scheme," *Design Codes and Cryptography*, vol. 38, no. 2, pp. 237–257, Feb. 2006.

[29] Anil K. Jain, Karthik Nandakumar, and Abhishek Nagar, *Security and Privacy in Biometrics*, chapter Fingerprint Template Protection: From Theory to Practice, pp. 187–214, Springer London, London, 2013.

[30] W. J. Scheirer and T. E. Boult, "Cracking fuzzy vaults and biometric encryption," in *Biometrics Symposium, 2007*, Sept 2007, pp. 1–6.

[31] Ruud M. Bolle, Jonathan H. Connell, and Nalini K. Ratha, "Biometric perils and patches," *Pattern Recognition*, vol. 35, pp. 2727–2738, 2002.

[32] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 846–859, May 2000.

[33] Rima Belguechi, Estelle Cherrier, Christophe Rosenberger, and Samy Ait-Aoudia, "Operational bio-hash to preserve privacy of fingerprint minutiae templates," *IET Biometrics*, vol. 2, no. 2, pp. 76–84, 2013.

[34] M. Ferrara, D. Maltoni, and R. Cappelli, "Noninvertible minutia cylinder-code representation," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1727–1737, 2012.

[35] Yagiz Sutcu, Shantanu Rane, Jonathan S. Yedidia, Stark C. Draper, and Anthony Vetro, "Feature extraction for a Slepian-Wolf biometric system using LDPC codes," in *IEEE International Symposium on Information Theory, ISIT 2008, Toronto, ON, Canada, July 6-11, 2008*, Frank R. Kschischang and En-Hui Yang, Eds. 2008, pp. 2297–2301, IEEE.

[36] Abhishek Nagar, Shantanu Rane, and Anthony Vetro, "Privacy and security of features extracted from minutiae aggregates," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2010, 14-19 March 2010, Dallas, Texas, USA*, 2010, pp. 1826–1829.

[37] Karthik Nandakumar and Anil K. Jain, "Biometric template protection: Bridging the performance gap between theory and practice," *IEEE Signal Processing Magazine*, vol. 32, no. 5, pp. 88–100, 2015.

[38] Julien Bringer and Vincent Despiegel, "Binary feature vector fingerprint representation from minutiae vicinities," in *Fourth IEEE International Conference on Biometrics: Theory Applications and Systems, BTAS 2010, Washington, DC, USA, 27-29 September, 2010*, 2010, pp. 1–6.

[39] Julien Bringer, Vincent Despiegel, and Melanie Favre, "Adding localization information in a fingerprint binary feature vector representation," in *Proceedings of the SPIE 8029, Sensing Technologies for Global Health, Military Medicine, Disaster Response, and Environmental Monitoring; and Biometric Technology for Human Identification VIII*, 2011, vol. 8029, p. 80291O.

[40] Berkay Topcu, Hakan Erdogan, Cagatay Karabat, and Berrin Yanikoglu, "Biohashing with fingerprint spectral minutiae," *Lecture Notes in Informatics*, vol. P-212, pp. 305–312, 2013.

[41] C. Karabat and Hakan Erdogan, "A cancelable biometric hashing for secure biometric verification system," in *Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Sept 2009, pp. 1082–1085.

[42] Zhengyao Bai and D. Hatzinakos, "LBP-based biometric hashing scheme for human authentication," in *11th International Conference on Control Automation Robotics Vision (ICARCV)*, Dec 2010, pp. 1842–1847.

[43] Yip Wai Kuan, Andrew Beng Jin Teoh, and David Chek Ling Ngo, "Secure hashing of dynamic hand signatures using Wavelet-Fourier compression with BioPhasor mixing and 2N discretization," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, pp. 32–32, 2007.

[44] C. Rathgeb and A. Uhl, “Iris-biometric hash generation for biometric database indexing,” in *20th International Conference on Pattern Recognition (ICPR)*, Aug 2010, pp. 2848–2851.

[45] Alessandra Lumini and Loris Nanni, “An improved biohashing for human authentication,” *Pattern Recognition*, vol. 40, no. 3, pp. 1057 – 1065, 2007.

[46] Matthew Turk and Alex Pentland, “Eigenfaces for recognition,” *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, Jan. 1991.

[47] P. T. Boufounos and S. Rane, “Secure binary embeddings for privacy preserving nearest neighbors,” in *Proceedings of Workshop on Information Forensics and Security (WIFS)*, Brazil, November 29 - December 2 2011.

[48] A. Adler, R. Youmaran, and S. Loyka, “Towards a measure of biometric information,” in *Canadian Conference on Electrical and Computer Engineering, 2006. CCECE ’06*, May 2006, pp. 210–213.

[49] Jovan Dj. Golic and Madalina Baltatu, “Entropy analysis and new constructions of biometric key generation systems.,” *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2026–2040, 2008.

[50] Cagatay Karabat and Berkay Topcu, “How to assess privacy preservation capability of biohashing methods?: Privacy metrics,” in *IEEE 22nd Signal Processing and Communications Applications Conference*, April 2014, pp. 2217–2220.

[51] Roman Viveros, K. Balasubramanian, and N. Balakrishnan, “Binomial and negative binomial analogues under correlated Bernoulli trials,” *The American Statistician*, vol. 48, no. 3, pp. 243–247, 1994.

[52] J. Ortega-Garcia, J. Fierrez, F. Alonso-Fernandez, J. Galbally, M.R. Freire, J. Gonzalez-Rodriguez, C. Garcia-Mateo, J.-L. Alba-Castro, E. Gonzalez-Agulla, E. Otero-Muras, S. Garcia-Salicetti, L. Allano, B. Ly-Van, B. Dorizzi, J. Kittler, T. Bourlai, N. Poh, F. Deravi, M. Ng, M. Fairhurst, J. Hennebert, A Humm, M. Tistarelli, L. Brodo, J. Richiardi, A Drygajlo, H. Ganster, F.M. Sukno, S.-K. Pavani, A Frangi, L. Akarun, and A Savran, “The multi-scenario multi-environment BioSecure Multimodal Database (BMDB),” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 6, pp. 1097–1111, June 2010.

[53] Paul Viola and Michael J. Jones, “Robust real-time face detection,” *International Journal on Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[54] Jean-Sebastien Coron and David Naccache, “An accurate evaluation of Maurer’s universal test,” in *Selected Areas in Cryptography*, vol. 1556 of *Lecture Notes in Computer Science*, pp. 57–71. Springer Berlin Heidelberg, 1999.

[55] Yaniv Plan and Roman Vershynin, “One-bit compressed sensing by linear programming,” *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1275–1297, 2013.

[56] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, “Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors,” *IEEE Transactions on Information Theory*, vol. 59, no. 4, April 2013.

[57] P.T. Boufounos and R.G. Baraniuk, “1-bit compressive sensing,” in *42nd Annual Conference on Information Sciences and Systems, CISS*, March 2008, pp. 16–21.

[58] E. J. Candes and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[59] Thomas Blumensath and Mike E. Davies, “Iterative hard thresholding for compressed sensing,” *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265 – 274, 2009.

[60] ISO-IEC, “Information technology – Biometric data interchange formats – Part 2: Finger minutiae data,” ISO 19794-2:2011, International Organization for Standardization, Geneva, Switzerland, 2011.

[61] Dario Maio, Davide Maltoni, Raffaele Cappelli, James L. Wayman, and Anil K. Jain, “FVC2002: Second fingerprint verification competition,” in *16th International Conference on Pattern Recognition, ICPR 2002, Quebec, Canada, August 11-15, 2002.*, 2002, pp. 811–814.

[62] Dario Maio, Davide Maltoni, Raffaele Cappelli, James L. Wayman, and Anil K. Jain, “FVC2002: Fingerprint Verification Competition,” `http://bias.csr.unibo.it/fvc2002/`, 2016, [Online; accessed 23-March-2016].

[63] Roy Wallace, Mitchell McLaren, Chris McCool, and Sébastien Marcel, “Intersession variability modelling and joint factor analysis for face authentication,” in

International Joint Conference on Biometrics, Los Alamitos, CA, USA, 2011, pp. 1–8.

[64] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[65] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[66] Simon Lucey and Tsuhan Chen, "A GMM parts based face representation for improved verification through relevance adaptation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, vol. 2, pp. 855–861.

[67] William M. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2002, May 13-17 2002, Orlando, Florida, USA*. 2002, pp. 161–164, IEEE.

[68] D. Maio, D. Maltoni, R. Cappelli, A. Franco, and M. Ferrara, "FVC-onGoing: on-line evaluation of fingerprint recognition algorithms," `https://biolab.csr.unibo.it/fvcongoing/UI/Form/Home.aspx`, 2016, [Online; accessed 23-March-2016].

[69] NIST Information Technology Laboratory, "Ongoing MINEX," `http://www.nist.gov/itl/iad/ig/ominex.cfm`, 2016, [Online; accessed 23-March-2016].

[70] NIST Information Technology Laboratory, "Fingerprint Vendor Technology Evaluation," `http://www.nist.gov/itl/iad/ig/fpvte2012.cfm`, 2012, [Online; accessed 23-March-2016].

[71] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM Journal on Computing*, vol. 38, no. 1, pp. 97–139, Mar. 2008.

[72] Cagatay Karabat, Mehmet Sabir Kiraz, Hakan Erdogan, and Erkay Savas, "THRIVE: threshold homomorphic encryption based secure and privacy preserving

biometric verification system," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 1, pp. 1–18, 2015.

[73] Man-Wai Mak and Wei Rao, "Acoustic vector resampling for GMMSVM-based speaker verification," in *Interspeech 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, 2010, pp. 1449–1452.

[74] Piotr Indyk and Rajeev Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, New York, NY, USA, 1998, STOC '98, pp. 604–613, ACM.

[75] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, New York, NY, USA, 2004, SCG '04, pp. 253–262, ACM.

[76] Michel X. Goemans and David P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the Association for Computing Machinery*, vol. 42, no. 6, pp. 1115–1145, Nov. 1995.

[77] Moses S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, New York, NY, USA, 2002, STOC '02, pp. 380–388, ACM.