IMPROVED SECURITY AND PRIVACY PRESERVATION FOR BIOMETRIC HASHING

by

ÇAĞATAY KARABAT

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

August 2013

IMPROVED SECURITY AND PRIVACY PRESERVATION FOR BIOMETRIC HASHING

APPROVED BY

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

Prof. Dr. Aytül ERÇİL .....

Assoc. Prof. Dr. Berrin YANIKOĞLU .....

Assoc. Prof. Dr. Erkay SAVAŞ .....

Asst. Prof. Dr. Engin MAŞAZADE .....

DATE OF APPROVAL: 02/08/2013 (2 August 2013)

To my son Emir, my wife Burçin and to my mother Sevgi...

Acknowledgements

This dissertation would not have been possible without the help of numerous people. First and foremost, I would like to express my sincere gratitude to Professor Hakan Erdogan for his invaluable help and encouragement during the preparation of this thesis. This thesis could not have been written without his guidance and patience. Next, I would like to thank my dissertation committee members, Prof. Erkay Savaş, Prof. Berrin Yanıkoğlu, Prof. Müjdat Çetin, Prof. Aytül Erçil, and Prof. Engin Maşazade for their precious time and valuable suggestions for the work done in this dissertation. Additionally, I want to thank to my unit head Oktay Adalıer and for his understanding during heavy periods.

My endless thanks are to my beautiful family. I should confess that, my desire for academic studies and consequently this thesis was possible thanks to love and encouragement of my sweety boy Mustafa Emir Karabat, my beloved wife Burçin Çetin Karabat, and my lovely mother Sevgi Tezcan. Also, I want to thank to TUBITAK BILGEM which is my place of employment for supporting me via PhD permission policy throughout my PhD study.

IMPROVED SECURITY AND PRIVACY PRESERVATION FOR BIOMETRIC HASHING

Çağatay KARABAT

EE, Ph.D. Thesis, 2013

Thesis Supervisor: Hakan Erdoğan

Keywords: Biohash, Privacy, Security, Cryptographic Protocols, Homomorphic Encryption, and Threshold Encryption.

Abstract

We address improving verification performance, as well as security and privacy aspects of biohashing methods in this thesis. We propose various methods to increase the verification performance of the random projection based biohashing systems. First, we introduce a new biohashing method based on optimal linear transform which seeks to find a better projection matrix. Second, we propose another biohashing method based on a discriminative projection selection technique that selects the rows of the random projection matrix by using the Fisher criterion. Third, we introduce a new quantization method that attempts to optimize biohashes using the ideas from diversification of error-correcting output codes classifiers. Simulation results show that introduced methods improve the verification performance of biohashing.

We consider various security and privacy attack scenarios for biohashing methods. We propose new attack methods based on minimum l1 and l2 norm reconstructions. The results of these attacks show that biohashing is vulnerable to such attacks and better template protection methods are necessary. Therefore, we propose an identity verification system which has new enrollment and authentication protocols based on threshold homomorphic encryption. The system can be used with any biometric modality and feature extraction method whose output templates can be binarized, therefore it is not limited to biohashing. Our analysis shows that the introduced system is robust against most security and privacy attacks conceived in the literature. In addition, a straightforward implementation of its authentication protocol is sufficiently fast enough to be used in real applications.

BİYOMETRİK KIYIM İÇİN ARTTIRILMIŞ GÜVENLİK VE MAHREMİYET KORUMASI

IMPROVED SECURITY AND PRIVACY PRESERVATION FOR BIOMETRIC HASHING

ÇAĞATAY KARABAT

EE, Doktora Tezi, 2013

Tez Danışmanı: HAKAN ERDOĞAN

Anahtar Kelimeler: Biyometrik Kıyım, Güvenlik, Mahremiyet, Kriptografik Protokoller, Homomorfik Şifreleme, ve Eşik Şifreleme

Özet

Bu tezde biyometrik kıyım yöntemlerinin doğrulama performanslarının arttırılmasının yanısıra güvenlik ve mahremiyet boyutlarını da ele aldık. Rastgele izdüşümü tabanlı biyometrik kıyım yöntemlerinin doğrulama performanslarını arttırmak için çeşitli yöntemler önerdik. lk olarak, en iyi doğrusal dönüşüme dayalı daha iyi bir izdüşümü matrisi bulmaya çalışan yeni bir biyometrik kıyım yöntemi önerdik. kinci olarak, rastgele izdüşümü matrisinin satırlarını Fisher kriterine göre seçen ayrıştırıcı bir izdüşümü seçimi tekniğine dayalı biyometrik kıyım yöntemi önerdik. Üçüncü olarak, biyometrik kıyım dizilerini hata düzeltme çıkış kodları sınıflandırıcılarının çeşitlendirilmesi için kullanılan fikirleri kullanarak optimize etmeye çalışan yeni bir nicemleme yöntemi sunduk.

Biyometrik kıyım yöntemleri için çeşitli güvenlik ve mahremiyet saldırıları düşündük. En az l1 ve l2 ölçütü yeniden yapılandırmalarına dayalı yeni saldırı yöntemleri önerdik. Bu saldırıların sonuçları biyometrik kıyımın böyle saldırılara karşı kırılgan olduğunu ve daha iyi şablon koruma yöntemlerinin gerekli olduğunu göstermektedir. Bu yüzden, eşik homomorfik şifrelemeye dayalı yeni kayıt ve doğrulama protokolleri içeren bir kimlik doğrulama sistemi önerdik. Sistem, çıkış şablonları ikili sayı dizisi haline getirilebilen herhangi bir biyometrik tür ve öznitelik çıkarma yöntemi ile çalışabilir, böylece biyometrik kıyım ile sınırlı değildir. Yaptığımız analizler sunduğumuz sistemin literatürde düşünülmüş birçok güvenlik ve mahremiyet saldırılarına karşı dayanıklı olduğunu göstermektedir. Ek olarak, sistemin doğrulama protokolünün basit bir gerçeklenmesi gerçek hayat uygulanmalarında kullanılabilecek derece hızlıdır.

Contents

List of Figures

List of Tables

Abbreviations

BQ	**B**inary **Q**uantization
CMU	**C**arnegie **M**ellon **U**niversity
COA	**C**iphertext-**o**nly **A**ttack
DET	**D**etection **E**rror **T**rade-off
DoS	**D**enial **o**f **S**ervice
DWT	**D**iscrete **W**avelet **T**ransform
ECC	**E**rror **C**orrection **C**ode
ECOC	**E**rror **C**orrecting **O**utput **C**odes
EER	**E**qual **E**rror **R**ate
FAR	**F**alse **A**ccept **R**ate
FRR	**F**alse **R**eject **R**ate
GMM	**G**aussian **M**ixture **M**odel
GSS	**G**olden **S**ection **S**earch
GS	**G**ram **S**chmidt
i.i.d	**I**dentically and **I**ndepently **D**istributed
KPA	**K**nown **P**laintext **A**ttack
LDA	**L**inear **D**iscrimant **A**nalysis
MSE	**M**ean **S**quared **E**rror
M2VTS	**M**ulti **M**odal **V**erification for **T**eleservices and **S**ecurity applications
PCA	**P**rinciple **C**omponent **A**nalysis
RNG	**R**andom **N**umber **G**enerator
ROC	**R**eceiver **O**perating **C**haracteristics
RP	**R**andom **P**rojection
SSL	**S**ecure **S**ockets **L**ayer
XOR	e**X**clusive **OR**

Chapter 1

Introduction

1.1 Motivation

With the development of computers, Internet and its applications that require authentication, the number of passwords that users have increased enormously in the digital age. Thus, users cannot generate and remember sufficiently strong keys, that are difficult to guess, for various applications. An alternative approach depends on authentication using biometrics that use physiological and/or behavioral traits (e.g. face, fingerprint, iris) for verifying the identity of individuals [5, 6]. Recent years have seen increased usage of biometric verification systems in many applications. Public and commercial organizations invest on secure electronic authentication (e-authentication) systems to reliably verify identity of individuals. Biometrics is one of the rapidly emerging technologies for e-authentication systems [7]. It offers several advantages (i.e. no need to remember your password, user friendly and convenient, cannot be shared, unique characteristics of individuals) over the traditional password based authentication systems. In biometric authentication systems, an input biometric template is compared to the reference biometric template either stored in a database server or a smart card for verification. The reference biometric template is stored as plaintext in a database or a smart card in most such systems.

It is impossible to discuss biometrics without security and privacy issues [1, 8]. Biometrics, which are stored in a smart card or a central database, is under security and privacy risks due to increased number of attacks against identity management systems in recent years [1, 8–10]. These systems are deemed insecure and raise about security and privacy concerns [11, 12]. A

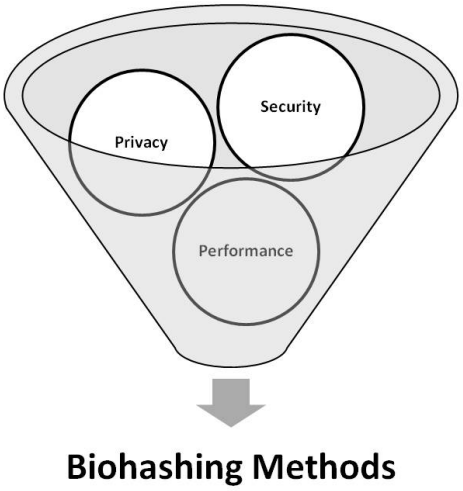

FIGURE 1.1: Three main aspects of biohashing methods.

proposed solution to handle aforementioned threats is to encrypt the reference biometric template stored in a smart card or a database by using cryptographic algorithms [13, 14]. The main problem of such solutions is that the encrypted reference biometric template must be decrypted to compare it with the claimer's input biometric template. This makes the systems weak against possible attacks at the verification stage.

Cancelable biometrics that combine the biometric with a secret key to enable randomized biometric hashing is a promising solution to cope with such problems [2, 15–17]. Biohashing schemes are one of the emerging biometric template protection methods [16, 18–21]. These methods offer low error rates and fast verification at the authentication stage. However, they suffer from several attacks reported in the literature [17, 22–24]. These schemes should be improved in order to be safely used in a wide range of real life applications.

In this thesis, we address three main aspects of biohashing methods as illustrated in Figure 1.1. These are

1. Performance aspects
2. Security aspects
3. Privacy aspects

First, we propose new biohashing methods in order to improve the verification performance of the existing random projection based biohashing methods. There are three main phases in a

biohashing method: 1) Feature extraction, 2)Dimension reduction, and 3)Quantization. We try to improve the verification performance by proposing new techniques in the dimension reduction and the quantization phases which have a large effect on the verification errors. We also take into account the key-stolen scenario where an attacker acquires the secret key of a legitimate user because if we assume that the key is always unknown, there would be no need for biometrics since it would be impossible to break into a system. The additional benefit of biometrics needs to be quantified. Our proposed methods have superior performance in comparison with the existing methods.

In addition to that, we address security and privacy aspects of the biohashing schemes. Although it is stated that random projection based biohashing methods satisfy irreversibility and cancelability property, we demonstrate that they cannot guarantee to satisfy these properties under some circumstances. We define some attack scenarios and perform them against a random projection based biohashing method in order to demonstrate security threats. For privacy threats, we focus on testing the irreversibility property of a biohash vector and try to obtain the biometric data under certain conditions.

Finally, we propose a new biometric verification system in order to cope with security and privacy flaws of the biohashing methods. The proposed system can also be seen as a new biometric template protection method. The proposed system includes novel enrollment and authentication protocols based on a threshold homomorphic cryptosystem in which the private key is shared between the user and the verifier. The system is designed for the malicious attack model where neither of the parties is assumed to be honest. Security of the system is enhanced using a two factor authentication scheme involving the users private key and the biometric data. In the proposed system, only encrypted binary biometric templates are stored in the database and verification is performed via homomorphically randomized templates, hence, original templates are never revealed even during authentication. Since threshold homomorphic encryption scheme is used, a malicious party cannot perform decryption on encrypted templates of the users in the database using a single key.

1.2 Contributions

In this thesis, we address verification performance and security and privacy preservation aspects of biohashing schemes. First, we develop new biohashing schemes in order to increase the verification performance even under the key-stolen scenario. Then, we analyze security and privacy gaps of the existing biohashing schemes and we discuss some possible attacks. Then, we develop a new biometric authentication system by taking into account previous attacks. Consequently, The contributions of this work can be summarized as follows:

1. We develop a new face image hashing method based on an optimal linear transformation [25]. In the proposed method, first, we apply a feature extraction method. Then, we define an optimal linear transformation matrix based on within-class covariance matrix which is the maximum likelihood estimate of the variations of the biometric data belonging to the same user. Next, we reduce the dimension of the feature vector by using this transform. Finally, we apply quantization and obtain a face image hash vector. We test the performance of the proposed method with various face databases and show that it has better performance even under the key-stolen scenario in comparison with the random projection (RP) based biohashing methods in the literature.

2. We develop a new biohashing scheme whose title is "Discriminative Projection Selection Based Face Image Hashing" [26]. In this work, we improve the performance of the random projection (RP) based biohashing schemes. The proposed method selects the rows of an RP matrix, which is a user dependent dimension reduction matrix, by using the Fisher criterion [27]. We also employ Gaussian mixture model (GMM) at the quantization step to obtain more distinct face image hash vectors for each user. The proposed method has better performance even under the key-stolen scenario in comparison with the RP based biohashing methods in the literature.

3. We develop a new biohashing scheme whose title is "Error-Correcting Output Codes Guided Quantization For Biometric Hashing" [28]. In this work, we improve the performance of the RP based biohashing schemes by introducing a new quantization method that attempts to optimize biometric hash vectors by using some ideas from Error-Correcting Output Codes (ECOC) classifiers. The proposed scheme shows superior performance even under the key-stolen scenario.

4. We analyze security and privacy gaps of the biohashing schemes. We perform irreversibility attacks and show that these attacks can threaten the privacy of the users. We also demonstrate that these attacks can threaten the security of the system since they allow an adversary to gain access with a high probability.

5. We develop a novel biometric authentication system which we call "THRIVE: Threshold Homomorphic encRyption based secure and privacy preserving bIometric VErification system" by taking into account the attacks against biohashing schemes [29]. It can be used in the applications where the user does not trust the verifier since the user does not need to reveal her biometric template and/or private key in order to authenticate herself and the verifier does not need to reveal any data to the user at the proposed authentication protocol. It is a two-factor authentication system (biometric and secret key) and is secure against illegal authentication attempts. In other words, a malicious adversary cannot gain access to the proposed system without having the biometric data and the private key of a legitimate user by performing adversary attacks described in [4] as well as hill-climbing attacks [30–33]. In the THRIVE system, the generated protected biometric templates are irreversible since they are encrypted.The proposed THRIVE system is developed in the malicious model and can be used with any existing biometric modality whose output can be binarized (not only with biohashing schemes). The THRIVE system lets only a legitimate user to enroll since signature scheme is used at the proposed enrollment stage. It is a new and advanced biometric template protection method without any helper data and only encrypted versions of binary templates are stored in the database and they are never released even during authentication. The THRIVE system also offers high level security and privacy features i.e. even if an adversary gains an access to the database and steals encrypted biometric templates, neither he can authenticate himself by using these encrypted biometric templates due to the authentication protocol nor he can decrypt these encrypted biometric templates due to the $(2, 2)$-threshold homomorphic encryption scheme. Furthermore, neither the verifier nor the user can perform decryption by themselves on encrypted biometric templates since the $(2, 2)$-threshold homomorphic encryption scheme is used. Instead, the verifier and the user can perform decryption collaboratively using their own private key shares. The verifier does not need to know the user's biometric template or private key in order to authenticate the user. In this system, authentication is performed via randomized templates which ensures privacy. Even if an adversary intercepts the

communication channel between the user and the verifier, he cannot obtain any useful information on the biometric template since all exchanged messages are randomized and/or encrypted and he cannot perform decryption due to the $(2, 2)$-threshold homomorphic encryption scheme. Furthermore, he cannot use the obtained data from message exchanges in this communication channel since nonce and signature schemes are used together in the authentication. In the THRIVE system, the generated protected biometric templates are cancelable. Even if they are stolen, they can be re-generated. It can also generate a number of protected templates from the same biometric data of a user due to the randomized encryption and biohashing. Thus, it ensures diversity. It is implemented and a successful authentication protocol run requires 0.218 seconds on average. Consequently, the proposed system is sufficiently efficient to be used in real world applications.

1.3 Thesis Organization

The thesis is structured as follows. Chapter 2 focuses on the basic background for the biohashing methods. Chapter 3, Chapter 4, and Chapter 5 address our works on performance improvement of biohashing methods. Chapter 3 explains a face image hashing method based on optimal linear transform under colored Gaussian noise assumption [25]. Chapter 4 introduces discriminative projection selection based face image hashing [26]. Chapter 5 is devoted to error-correcting output codes guided quantization for biometric hashing [28].

In addition to these works, security and privacy aspects of biohashing methods are covered in Chapter 6 and Chapter 7. We address security and privacy attacks against biohashing methods in Chapter 6. Finally, we propose a novel biometric verification system called "THRIVE" in Chapter 7 by taking into account the security flaws and privacy threats in the previous chapter [29]. Finally, we conclude the thesis and discuss the future work in Chapter 8.

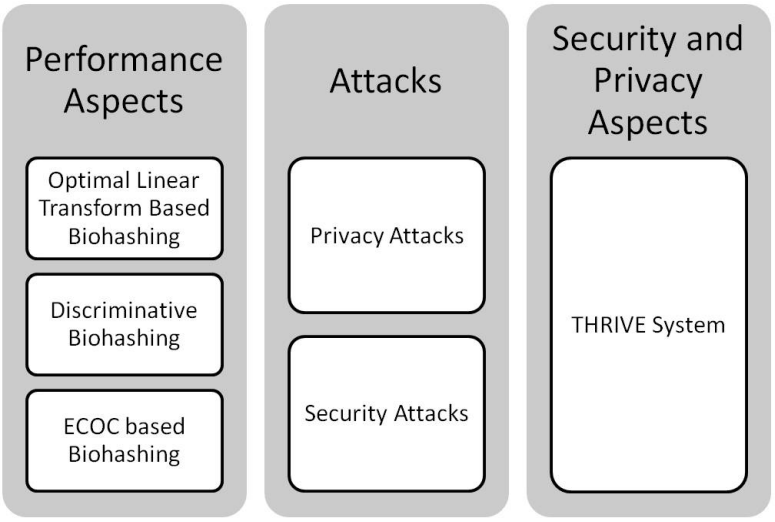

FIGURE 1.2: Classification of the work that has been performed in this thesis.

Chapter 2

Background

2.1 Preliminaries

2.1.1 Biohashing Based Verification System

In recent years, biohashing is one of the emerging biometric template protection methods in the literature [16, 18–21]. Biohash is a binary and pseudo-random representation of a biometric template. Biohashing methods use two inputs: 1) Biometric template, 2) User's secret key. A biometric feature vector is transformed into a lower dimensional sub-space using a pseudo-random set of orthogonal vectors which are generated from the user's secret key. Then, the result is binarized to produce a bit-string which is called the biohash. In an ideal case, the distance between the biohashes belonging to the 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 to achieve lower false acceptance rates. The desired properties of the biohashes are summarized as follows:

1. The biohash should be irreversible so that biometric template cannot be obtained from a biohash vector.

2. The biohash should be cancelable so that it can be renewed when an attacker steals it.

3. The biohash should be robust against different biometric images belonging to the same user so that the Hamming distance between the biohash vectors (i.e. generated from the

same secret key but different biometric image collected at different session) of the same user should be small.

4. Biohash should be fragile to the biometric images which do not belong to the same legitimate user so that the Hamming distance between the biohash vectors (i.e. generated from different secret key and different biometric image) of the different users should be high.

Biohashing based verification systems perform an automatic verification of a user based on her specific biometric data and secret key. There are two main stages in these systems:

1. Enrollment stage,
2. Authentication stage.

The user is enrolled to the system at the enrollment stage. Then, the user again provides her biometric data to the system at the authentication stage in order to prove her identity. Biohashing schemes are simple yet powerful biometric template protection methods [16, 18–21]. In this part, we describe the random projection (RP) based biohashing scheme proposed by Ngo *et al.* [2]. In a RP based biohashing method, there are three main phases in each stage and these phases are described as follows:

1. Feature extraction,
2. Dimension reduction,
3. Quantization.

These three phases for the face biometric are explained in the following.

2.1.1.1 Enrollment Stage

In the enrollment stage, a user enrolls to the biometric verification system by giving her face image and secret key to the system. Then, the system computes her biohash and stores it for verification purposes at the authentication system.

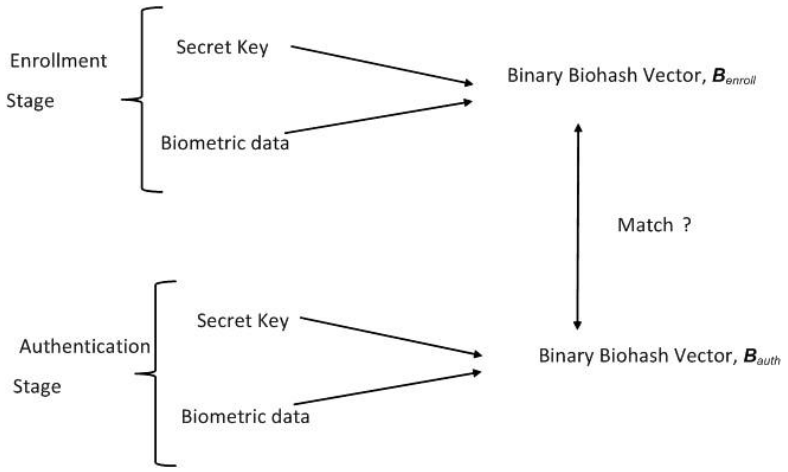

FIGURE 2.1: Illustration of biohashing based verification.

Feature Extraction At this phase, a user gives her face image, $\mathbf{I}_{enroll} \in \mathbb{R}^{m \times n}$, to the system. The face image is lexicographically re-ordered and the face vector, $\mathbf{x}_{enroll} \in \mathbb{R}^{(mn)\times 1}$, is obtained. Then, principle component analysis [34] is applied to it for feature extraction as follows:

$$\boldsymbol{y}_{enroll} = \mathbf{A}(\boldsymbol{x}_{enroll} - \boldsymbol{\mu}), \tag{2.1}$$

where $\mathbf{A} \in \mathbb{R}^{k\times(mn)}$ is the pre-computed PCA matrix trained by the face images in the training set, $\boldsymbol{\mu}$ is the pre-computed mean face vector by the face images in the training set, and $\boldsymbol{y}_{enroll} \in \mathbb{R}^{k\times 1}$ is the vector containing PCA coefficients belonging to the user.

Dimension Reduction At this phase, a RP matrix, $\mathbf{R} \in \mathbb{R}^{\ell\times k}$, is generated to reduce the dimension of 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 Random Number Generator (RNG) with a seed derived from the user's secret key. The Gram-Schmidt (GS) procedure is applied to obtain an orthonormal projection matrix $\mathbf{R}_{GS} \in \mathbb{R}^{\ell\times k}$ to have more distinct projections. Finally, PCA coefficients are projected onto a lower ℓ-dimensional subspace as follows:

$$\boldsymbol{z}_{enroll} = \mathbf{R}_{GS}\boldsymbol{y}_{enroll}, \tag{2.2}$$

where $\boldsymbol{z}_{enroll} \in \mathbb{R}^{\ell\times 1}$ is the intermediate biohash vector belonging to the user.

2.1.1.2 Quantization

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

$$\boldsymbol{B}_{enroll}(i) = \begin{cases} 1 & \text{if} \quad z_{enroll}(i) \geq \beta, \\ 0 & \text{Otherwise,} \end{cases} \tag{2.3}$$

where $i = 1, \ldots, \ell$, $\boldsymbol{B}_{enroll} \in \{0,1\}^{\ell}$ denotes biohash vector of the user and β is the mean value of the intermediate biohash vector $\mathbf{z}_{enroll}$.
The computed binary biohashes are stored in the database in the enrollment stage for verification purpose during the authentication stage.

2.1.1.3 Authentication Stage

In the authentication stage, exactly same operations are performed on the biometric face image supplied by the user. The user is authenticated when the Hamming distance between $\boldsymbol{B}_{enroll}$ (which denotes the biohash of the user generated at the enrollment stage) and $\boldsymbol{B}_{auth}$ (which denotes the biohash of the user generated at the authentication stage) is below a distance threshold t as follows:

$$d(\boldsymbol{B}_{enroll}, \boldsymbol{B}_{auth}) = \sum_{i=1}^{\ell} \boldsymbol{B}_{enroll}(i) \oplus \boldsymbol{B}_{auth}(i) \leq t, \tag{2.4}$$

where $d(\boldsymbol{B}_{enroll}, \boldsymbol{B}_{auth})$ denotes the Hamming distance between $\boldsymbol{B}_{enroll}$ and $\boldsymbol{B}_{auth}$, $\oplus$ denotes the binary XOR (exclusive OR) operator and t denotes the decision threshold. Therefore, the verifier decides whether the user is legitimate or not using the decision threshold.

2.1.2 Performance Measures for Biometric Verification

In a biometric verification system, a user must first claim that he/she is someone who has been enrolled into the system, and the system then determines if the users claim is true or false. The biometric verification system makes a decision by using the below decision function:

$$decision = \begin{cases} \text{accept} & \text{if } d(\boldsymbol{B}_{enroll}, \boldsymbol{B}_{auth}) < t, \\ \text{reject} & \text{Otherwise,} \end{cases} \tag{2.5}$$

where $d(\boldsymbol{B}_{enroll}, \boldsymbol{B}_{auth})$ denotes the Hamming distance between the biohashes computed at the enrollment and the authentication stages as in Eq. 2.4 and t denotes the decision threshold.

In this part, we describe performance measures for biometric verification used in this thesis. The verification performance of biometric systems is usually expressed in terms of their False Acceptance Rate (FAR), False Rejection Rate (FRR) and the related Equal Error Rate (EER). In addition to these metrics, there are some performance charts like detection error tradeoff (DET) graph which plots FRR versus FAR [35]. These metrics and charts are used for reflecting the system performance.

The biometric verification systems may make two types of errors due to the accept/reject outcomes i.e., false acceptance (FA) and false rejection (FR). For biohashing based verification systems, FAR is an empirical estimate of the probability (the percentage of times) at which the system incorrectly accepts a biohash of the claimer when the biohash actually belongs to a different user (impostor). In other words, it is the case where the system falsely accepts the claim although the actual claimer is an impostor. On the other hand, FRR is an empirical estimate of the probability (the percentage of times) at which the system incorrectly rejects a biohash of the claimer when the biohash actually belongs to the genuine user. In other words, it is the case where the system falsely rejects a genuine users claim. The FAR and FRR of the corresponding system can be estimated in the following ways:

$$\text{FRR}(t) = \frac{\text{FA}(t)}{N^g}, \tag{2.6}$$

and

$$\text{FAR}(t) = \frac{\text{FR}(t)}{N^i}, \tag{2.7}$$

where FA and FR count the number of FA and FR accesses respectively; and N^g and N^i denote the total number of genuine and imposter accesses respectively.

FAR and FRR curves can be plotted as a function of the decision threshold. The FAR is a monotonically increasing function of the decision threshold whereas the FRR is a monotonically

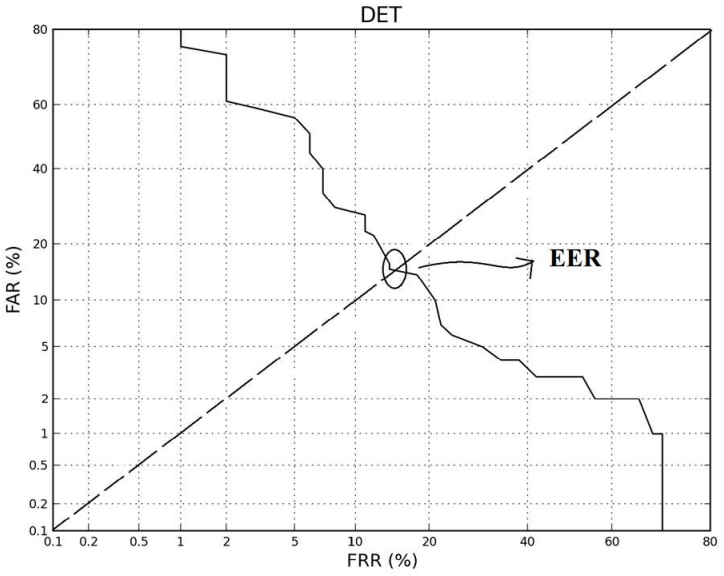

FIGURE 2.2: Illustration of a DET curve. Each point on a DET curve corresponds to a specific threshold value although threshold values are not evident from the curve. EER can be found from the intersection of the DET curve with a straight line hugging the left and the top borders.

decreasing function of the decision threshold. Therefore, it is impossible to minimize the two error rates simultaneously. EER is related with the FAR and the FRR. It is the rate at which the FAR is equal to the FRR for a certain threshold t_e.

$$\text{EER} = \text{FRR}(t_e) = \text{FAR}(t_e). \tag{2.8}$$

The DET curve plots FRR versus FAR for all possible values of the threshold t but the axes are often scaled non-linearly to highlight the region of error rates of interest. Commonly used scales include normal deviate scale and logarithmic scale. It is similar to receiver operating characteristics (ROC) curve (which plots probability of correct acceptance (1-FRR) in the Y-axis versus FAR in the X-axis) except that the axes are often scaled non-linearly to highlight the region of error rates of interest. An example DET curve can be seen in Figure 2.2.

2.1.3 Principle Component Analysis (PCA)

PCA is one of the most common feature extraction techniques which is used for face images in the literature [34]. PCA can be used in a number of applications e.g. face recognition, data compression. PCA is the optimum linear dimensionality reduction technique with respect to

mean squared error (MSE) of the reconstruction for a given data set. The basic steps of the PCA is as follows:

1. We are given a set of M training face images $\mathbf{I}_i \in \mathbb{R}^{m \times n}$ where $i = 1, \ldots, M$. We lexicographically re-order them in order to obtain face vectors $\boldsymbol{x}_i \in \mathbb{R}^{K \times 1}$ where $K = m \times n$. We compute the sample mean, $\boldsymbol{\mu}$, of the face vectors as follows:

$$\boldsymbol{\mu} = \frac{1}{M} \sum_{i=1}^{K} \boldsymbol{x}_i. \tag{2.9}$$

   Then, we subtract the sample mean face image vector from the training face image vectors.

$$\boldsymbol{t}_i = \boldsymbol{x}_i - \boldsymbol{\mu}. \tag{2.10}$$

2. We compute covariance matrix $\mathbf{C}$ of the training face vectors

$$\mathbf{C} = \frac{1}{M} \sum_{i=1}^{M} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T = \frac{1}{M} \sum_{i=1}^{M} (\boldsymbol{t}_i)(\boldsymbol{t}_i)^T = \mathbf{B}\mathbf{B}^T, \tag{2.11}$$

   where $\mathbf{B} = [\boldsymbol{t}_1, \boldsymbol{t}_2, \cdots, \boldsymbol{t}_M] \in \mathbb{R}^{K \times M}$.

3. We want to compute eigenvalues λ_j's and eigenvectors of $\mathbf{C}$, however, computing the eigenvectors of $\mathbf{C}$ is not an easy task for typical face image sizes when $K \gg M$. Thus, we first compute the eigenvectors of the much-smaller $M \times M$ matrix $\mathbf{B}^\top\mathbf{B}$ in order to efficiently compute the eigenvectors, $\mathbf{U}$= $\{\mathbf{u}_1, \ldots, \mathbf{u}_L\}$, of $\mathbf{C}$. Here, the eigenvalues of $\mathbf{B}^\top\mathbf{B}$ and $\mathbf{B}\mathbf{B}^\top$ are the same whereas their eigenvectors are different. The singular value decomposition of $\mathbf{B}$ is as follows:

$$\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^T, \tag{2.12}$$

   where $\mathbf{U} \in \mathbb{R}^{K \times K}$ is a unitary matrix, $\Sigma \in \mathbb{R}^{K \times M}$ is a rectangular diagonal matrix with non-negative real numbers on the diagonal, and $\mathbf{V}^T \in \mathbb{R}^{M \times M}$ is a unitary matrix. The diagonal entries of Σ matrix are known as the singular values of $\mathbf{B}$. The eigendecomposition of $\mathbf{B}^\top\mathbf{B}$ and $\mathbf{B}\mathbf{B}^\top$ are as follows:

$$\mathbf{B}^T\mathbf{B} = \mathbf{V}\Sigma^2\mathbf{V}^T, \tag{2.13}$$

$$\mathbf{B}\mathbf{B}^T = \mathbf{U}\Sigma^2\mathbf{U}^T. \tag{2.14}$$

From Eq. 2.12 the eigenvectors of $\mathbf{B}\mathbf{B}^\top$ can be computed as follows:

$$\mathbf{U} = \mathbf{B}\mathbf{V}\Sigma^{-1}, \tag{2.15}$$

where diagonal entries of the Σ matrix contains the square root of the eigenvalues of $\mathbf{B}\mathbf{B}^\top$.

4. We define a projection matrix $\mathbf{A}$ composed of N eigenvectors of $\mathbf{C}$ with highest eigenvalues $\mathbf{U}$= $\{\mathbf{u}_1, \ldots, \mathbf{u}_N\}$ as follows:

$$\mathbf{A} = \begin{pmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_N^T \end{pmatrix}, \tag{2.16}$$

where $\mathbf{u}_1$ is the eigenvector of $\mathbf{C}$ with the highest eigenvalue.

5. Finally, we can compute the N-dimensional representation of the original K-dimensional face vector as follows:

$$\boldsymbol{y}_i = \mathbf{A}(\boldsymbol{x}_i - \boldsymbol{\mu}). \tag{2.17}$$

2.1.4 Random Number Generation

Pseudo random number generation is the process of generating a sequence of numbers using deterministic computations where an outside observer would consider the sequence as being randomly generated. The pseudo-random number generators require a seed value to start the computations and would generate exactly the same sequence of numbers if given the same seed.

Mersenne twister is a pseudo random number generator proposed by Makoto Matsumoto and Takuji Nishimura [36]. The 32-bit Mersenne twister algorithm produces uniformly random integers between 0 and $2^{32} - 1$ and its period is approximately 10^{6001}. The integer values can be normalized to generate what appears to be uniformly random real values between 0 and 1.

The *rand*(.) function in MATLAB generates a uniformly distributed pseudo-random number by using this algorithm. The *rand*('*state*', *s*) causes the *rand*(.) function to initialize the generator

with the seed s which is a scalar integer. The user's secret key is used as a seed in order to generate random numbers for the random projection matrix. We generate a matrix containing pseudo-random values drawn from the standard uniform distribution on the open interval $(0, 1)$. Let the random projection matrix be $\mathbf{R} \in \mathbb{R}^{\ell \times k}$ and let $r(i, j)$ denote the element located at the i^{th} row and j^{th} column of $\mathbf{R}$. In this case, the random variable $r(i, j)$ has the standard uniform distribution with minimum 0 and maximum 1.

2.2 Related Work

Security and privacy concerns on biometrics limit their widespread usage in real life applications. The initial solution that comes to mind for security and privacy problems is to use cryptographic primitives. On the other hand, biometric templates cannot be directly used with conventional encryption techniques (i.e. AES, 3DES) since biometric data are inherently noisy [37]. In other words, the user is not able to present exactly the same biometric data repeatedly. Namely, when a biometric template is encrypted during the enrollment stage, it should be decrypted to pass the authentication stage for comparison with the presented biometric. This, however, again leads to security and privacy issues for biometric templates at the authentication stage [37]. Another problem with regards to such a solution is the key management, i.e. storage of encryption keys. When a malicious database manager obtains encryption keys, he can perform decryption and obtain biometric templates of all users. Similar problems are valid for cryptographic hashing methods. Since cryptographic hash is a one-way function, when a single bit is changed the hash sum becomes completely different due to the avalanche effect [38]. Thus, successful authentication by exact matching cannot be performed even for legitimate users due to the noisy nature of biometric templates. Therefore, biometric templates also cannot directly be used with traditional cryptographic hashing methods.

Biometric systems which use error correction methods are proposed in order to cope with noisy nature of the biometric templates in the literature [39–41]. In such systems, after using error correction, the biometric data collected at the authentication stage can become exactly the same with the biometric data collected at the enrollment stage due to tolerance to a limited number of errors brought by the error correction methods. In other words, these systems can get error-free biometric templates and thus cryptographic primitives (i.e. encryption and hashing) can successfully be employed without suffering from the avalanche effect [13, 37, 41, 42]. However, large error correcting capability requirements makes them impractical for real life applications

[43]. Furthermore, side information (parity bits) is needed for error correction and this may lead to information leakage and even other attacks (i.e., error correcting code statistics, and non-randomness attacks) [44]. Besides, Zhou *et al.* clearly demonstrate in their work that redundancy in an error correction code causes privacy leakage for biometric systems [45].

Biohashing schemes are simple yet powerful biometric template protection methods [16, 18–21]. It is worth pointing out that biohashing is completely different from cryptographic hashing. In the literature, researchers propose various biometric hashing methods which mostly depend on random projections where the biometric template is projected over a set of randomly selected orthogonal vectors [2, 3, 16–18, 46]. They argue that even when an attacker steals the biometric hash vector, he cannot obtain the original biometric template. Thus, their scheme preserves privacy of the users. In these works, they propose a two factor authentication based on a user-defined password and a biometric template. The feature extraction phase of a biometric system is randomized by using iterated inner products between a tokenized pseudo-random vector and the user specific biometric features. The features may be generated from principal component analysis (PCA), Discrete Wavelet Transform (DWT) and Linear Discriminant Analysis (LDA) etc. Finally, they employ binary quantization to obtain face image hash vectors. Eventually, they produce a set of user specific biometric code that they called biometric hash or biohash.

There are various works on biohashing methods that uses different biometric modalities. Ngo *et al.* [2, 3] and Karabat *et al.* [26, 28] propose random projection based biohashing methods for face images whereas Lumini *et al.* [16] work on fingerprint based biohashing methods. On the other hand, Vielhauer *et al.* [47] develop a biohashing method based on statistical features in online signatures. Connie *et al.* [48] develop a biohashing method for palmprints. In addition to these works, there are other biohashing methods which works with multimodal biometrics. For instance, Fuksis *et al.*[49] propose a biohashing method based on fusion of data coming from palmprint and palm vein.

Although biohashing schemes are proposed to solve security and privacy issues, there are still security and privacy issues associated with them [16, 17, 22–24]. Lumini *et al.* [16] report that when the secret keys are compromised, biohashing methods cannot achieve near zero equal error rate (EER) and they show that this assumption is unrealistic. They propose a key-stolen attack scenario and they investigate the performance of the random projection based biohashing methods when an attacker gets the secret key of a user. In addition to that, other researchers claim that biohashes can be reversible under certain conditions and an adversary can estimate

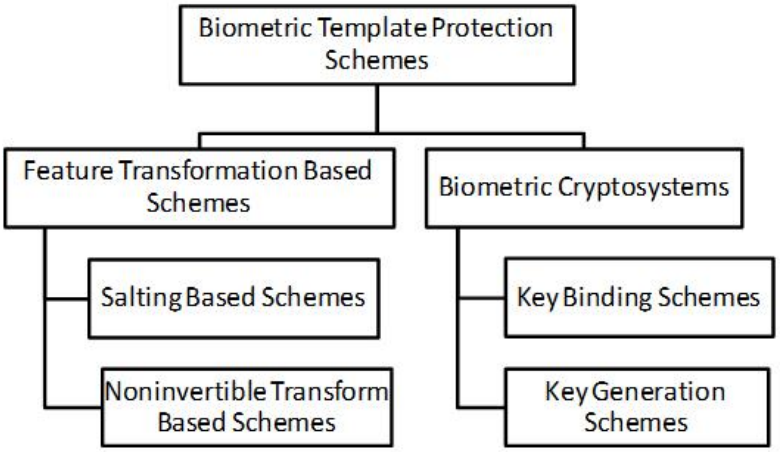

FIGURE 2.3: General classification of biometric template protection schemes (adapted from [1]).

biometric template of a user from her biohash [17, 22–24]. Consequently, when biohashes are stored in the databases and/or smart cards in their plain form, they can threaten the security of the system as well as the privacy of the users. Moreover, an adversary can use an obtained biohash in order to threaten the system security by performing malicious authentication. When the secret key is compromised, an adversary may reconstruct a biometric template that resembles the original template even though an inversion which would yield the exact template may not be possible. Thus, these schemes are considered as "generally invertible" in some publications [1].

In the literature, Jain *et al.* classify biometric template protection schemes into two main categories [1]: 1) Feature transformation based schemes, 2) Biometric cryptosystems as illustrated in Figure 2.3. Although biometric template protection methods are proposed to overcome security and privacy problems of biometrics [1, 18–21, 26, 28, 50–58], recent research shows that security and privacy issues still persist for these schemes [16, 17, 22–24, 59–61]. Furthermore, there are a number of works on privacy leakages of biometric template protection methods in the literature [45, 62–65]. Zhou *et al.* propose a framework for security and privacy assessment of biometric template protection methods [45]. Ignatenko *et al.* analyze the privacy leakage in terms of the mutual information between the public helper data and biometric features in a biometric template protection method. A trade-off between maximum secret key rate and privacy leakage is given in their works [63, 66].

The main idea behind biometric cryptosystems (also known as biometric encryption systems) is either binding a cryptographic key with a biometric template or generating the cryptographic key directly from the biometric template [67]. Thus, the biometric cryptosystems can be classified

into two main categories: 1) Key binding schemes, 2) Key generation schemes. Biometric cryptosystems use helper data, which is public information, about the biometric template for verification. Although helper data are supposed not to leak any critical information about the biometric template, Rathgeb *et al.* show that helper data is vulnerable to statistical attacks [68]. Furthermore, Ignatenko *et al.* show how to compute a bound on possible secret rate and privacy leakage rate for helper data schemes [69]. Adler performs hill-climbing attack against biometric encryption systems [60]. Besides, Stoianov *et al.* propose several attacks (i.e., nearest impostors, error correcting code statistics, and non-randomness attacks) to biometric encryption systems [44].

In the literature, fuzzy commitment [41] and fuzzy vault schemes [58] are categorized under the key binding schemes. These schemes aim to bind a cryptographic key with a biometric template. In ideal conditions, it is infeasible to recover either the biometric template or the random bit string without any knowledge of the user's biometric data. However, this is not the case in reality because biometric templates are not uniformly random. Furthermore, error correction codes (ECC) used in biometric cryptosystems lead to statistical attacks (i.e., running ECC in a soft decoding or erasure mode and ECC Histogram attack) [44, 70]. Ignatenko *et al.* show that fuzzy commitment schemes leak information in cryptographic keys and biometric templates which leads to security flaws and privacy concerns [63, 66]. In addition, Zhou *et al.* argue that fuzzy commitment schemes leak private data. Chang *et al.* describe a non-randomness attack against the fuzzy vault scheme which causes distinction between the minutiae points and the chaff points [71]. Moreover, Kholmatov *et al.* perform a correlation attack against the fuzzy vault schemes [72].

Keys are generated from helper data and a given biometric template in key generation schemes [1]. Fuzzy key extraction schemes are classified under key generation schemes and use helper data [73–77]. These schemes can be used as an authentication mechanism where a user is verified by using her own biometric template as a key. Although fuzzy key extraction schemes provide key generation from biometric templates, repeatability of the generated key (in other words stability) and the randomness of the generated keys (in other words entropy) are two major problems of them [1]. Boyen *et al.* describe several vulnerabilities (i.e. improper fuzzy sketch constructions may leak information on the secret, biased codes may cause majority vote attack, and permutation leaks) of the fuzzy key extraction schemes from outsider and insider attacker perspectives [78]. Moreover, Li *et al.* mention that when an adversary obtains sketches, they may reveal the identity of the users [79].

Biohashing schemes are simple yet powerful biometric template protection methods [16, 18–21] which can be classified under salting based schemes. It is worth pointing out that biohashing is completely different from cryptographic hashing. Although biohashing schemes are proposed to solve security and privacy issues, there are still security and privacy issues associated with them [17, 22–24]. In these works, the authors claim that biohashes can be reversible under certain conditions and an adversary can estimate biometric template of a user from her biohash. Consequently, when biohashes are stored in the databases and/or smart cards in their plain form, they can threaten the security of the system as well as the privacy of the users. Moreover, an adversary can use an obtained biohash in order to threaten the system security by performing malicious authentication.

Non-invertible transform based schemes use a non-invertible transformation function, which is a one-way function, to make the biometric template secure [80–82]. User's secret key determines the parameters of non-invertible transformation function and this secret key should be provided at the authentication stage. Even if an adversary obtains the secret key and/or the transformed biometric template, it is computationally hard to recover the original biometric template. On the other hand, these schemes suffer from the trade-off between discriminability and non-invertibility which limits their recognition performance [1].

Apart from the aforementioned schemes, another approach is the use of cryptographic primitives (i.e. encryption, hashing) to protect biometric templates. These works generally focus on fingerprint-based biometric systems. Tuyls *et al.* propose a fingerprint authentication system which incorporates cryptographic hashes [83]. They use an error correction scheme to get exactly the same biometric template from the same user in each session which is similar to the fuzzy key extraction schemes. They store cryptographic hashes of biometric templates in the database and make comparison in the hash domain. However, there is no guarantee to get exactly the same biometric templates from the user even if the system incorporates an error correction scheme in real life applications since it is limited with the pre-defined threshold of error correction capacity. They also use helper data which are sent over a public channel and this may lead to security flaws as well. Apart from that, an adversary can threaten the security of the system when he performs an attack against the database since he can obtain the user id, helper data and the hashed version of the secret which is generated by the biometric data and the helper data. Although the adversary cannot obtain the biometric data itself in its plain form, he can get all needed credentials (i.e. hash values of the secrets) in order to gain access to the system.

Nowadays, homomorphic encryption methods are used with biometric feature extraction methods in order to perform verification via encrypted biometric templates [54, 84–86]. These methods, however, offer solutions in the honest-but-curious model where each party is obliged to follow the protocol but can arbitrarily analyze the knowledge that it learns during the execution of the protocol in order to obtain some additional information. Their proposed system is not designed for the malicious model where each party can arbitrarily deviate from the protocol and may be corrupted. On the other hand, they do not take into account security and privacy issues of biometric templates stored in the database [54, 86]. The authors state that their security model will be improved in the future work by applying encryption methods also on the biometric templates stored in the database. Furthermore, some of these systems are just designed for a single biometric modality or a specific feature extraction method which limits their application areas [84, 85]. Apart from that, an adversary can enroll himself on behalf of any user to their systems since they do not offer any solutions for malicious enrollment. Finally, all these systems suffer from computational complexity.

Kerschbaum *et al.* propose a protocol in order to compare fingerprint templates without actually exchanging them by using secure multi-party computation in the honest-but-curious model [87]. At the enrollment stage, the user gives her fingerprint template, minutiae pairs and PIN to the system. Thus, the verifier knows the fingerprint templates which are collected at the enrollment stage. Although the user does not send her biometric data at the authentication, the verifier already has the user's enrolled biometric data and this threatens the privacy of the user in case of a malicious verifier. In addition to that, a malicious verifier can use these fingerprint templates for malicious authentication. Furthermore, since the fingerprint comparison reveals the matching scores (i.e. Hamming distance), the attacker can perform a hill climbing attack against this system. Apart from these security and privacy flaws, the authors just focus on secure comparison in their protocol and they do not develop any solutions for the malicious model.

Erkin *et al.* [84] propose a privacy preserving face recognition system for the eigen-face recognition algorithm [34]. They design a protocol that performs operations on encrypted images by using the Pailler homomorphic encryption scheme. Then, Sadeghi *et al.* improve the efficiency of this system [85]. In their work, they merge the eigen-face recognition algorithm with homomorphic encryption schemes. However, they limit the recognition performance of the system with the eigen-face method although there are various feature extraction methods which perform better than it. Unfortunately, their system cannot be used for any other feature extraction

method for face images. Moreover, they do not employ a threshold cryptosystem which prevents from a malicious party aiming to perform decryption by himself. Storing face images (or corresponding feature vectors) in the database in plain is the most serious security flaw of this system. An adversary, who gains access to the database, can obtain all face images. Therefore, the adversary can perform an attack against the database which definitely threatens the security of the system and the privacy of the users.

Barni *et al.* [54, 86] propose a privacy preservation system for fingercode templates by using homomorphic encryption in the honest-but-curious model. However, they do not propose any security and privacy solutions on the biometric templates stored in the database. This issue is mentioned as future work in their paper. In addition to that, they do not employ threshold encryption which would prevent from a malicious party aiming to perform decryption by himself. Therefore, their proposed system is open to adversary attacks against the database as stated in their work as well. They do not address the malicious enrollment issue as well. Moreover, the user should trust the server in their system. Although they achieve better performance than [84, 85] in terms of bandwidth saving and time efficiency, they do not address the applications where the user and the verifier do not trust each other (e.g. the malicious model).

Chapter 3

A Face Image Hashing Method Based on Optimal Linear Transform Under Colored Gaussian Noise Assumption

3.1 Introduction

[1] In this chapter, the aim is to find a better projection matrix in order to reduce the Hamming distance between the biometric hash vectors which represent the same user but differ due to variations in the biometric data. This projection matrix is found by using the optimal linear transform under colored Gaussian noise assumption [25]. In the literature, Mihcak *et al.* worked on the optimal dimension reduction problem for various digital communication problems in their work [88]. They have modeled the noise between the transmitter and receiver as additive colored Gaussian noise. The model also assumes the noise is independent of the source signal. Finally, they derive a formula for the set of optimal linear transforms to reduce the dimension of the data transmitted through this noisy channel minimizing the probability of error under the assumptions [88].

The additive colored Gaussian channel noise between the transmitter and receiver can be used to model the acquisition variability of the biometric images of the same user. In other words, the user gives different images to the system at each enrollment/authentication session. Thus,

[1] This chapter is based on [25].

there are a number of different face images belonging to the same user in the system. We assume that difference between the face images belonging to the same user can be modeled as additive colored Gaussian noise. We have adopted this approach to the biometric hashing methods and develop a new biometric hashing method which is based on the within-class covariance matrix. This optimal linear transformation enables us to better define the biometric face images in a reduced dimensional space. Thus, the proposed method improves the performance of the biometric verification systems in comparison with biometric hashing methods proposed in the literature that are based on random projections [2, 3, 16–18, 46]. The set of linear transforms derived in [88] still allows using random linear projections, so that the random nature of biohash extraction is also preserved in our approach.

The performance of the proposed face image hashing method increases with the increasing biometric feature vector length as well as the increasing biometric hash vector length. In other words, we get better performance with the higher dimension in biometric hash vector. It is obvious that the performance of the proposed method also depends on the database.

3.2 The Biometric Verification System Based on the Proposed Face Image Hashing Method

In this section, we introduce the proposed biometric verification system based on the proposed face image hashing method. In the proposed biometric verification method, there are two main stages:

1. Enrollment stage,
2. Authentication stage.

These stages are addressed in the below parts.

3.2.1 Enrollment Stage

In this part, we introduce the enrollment stage which consists of three main phases. These are

1. Feature extraction,

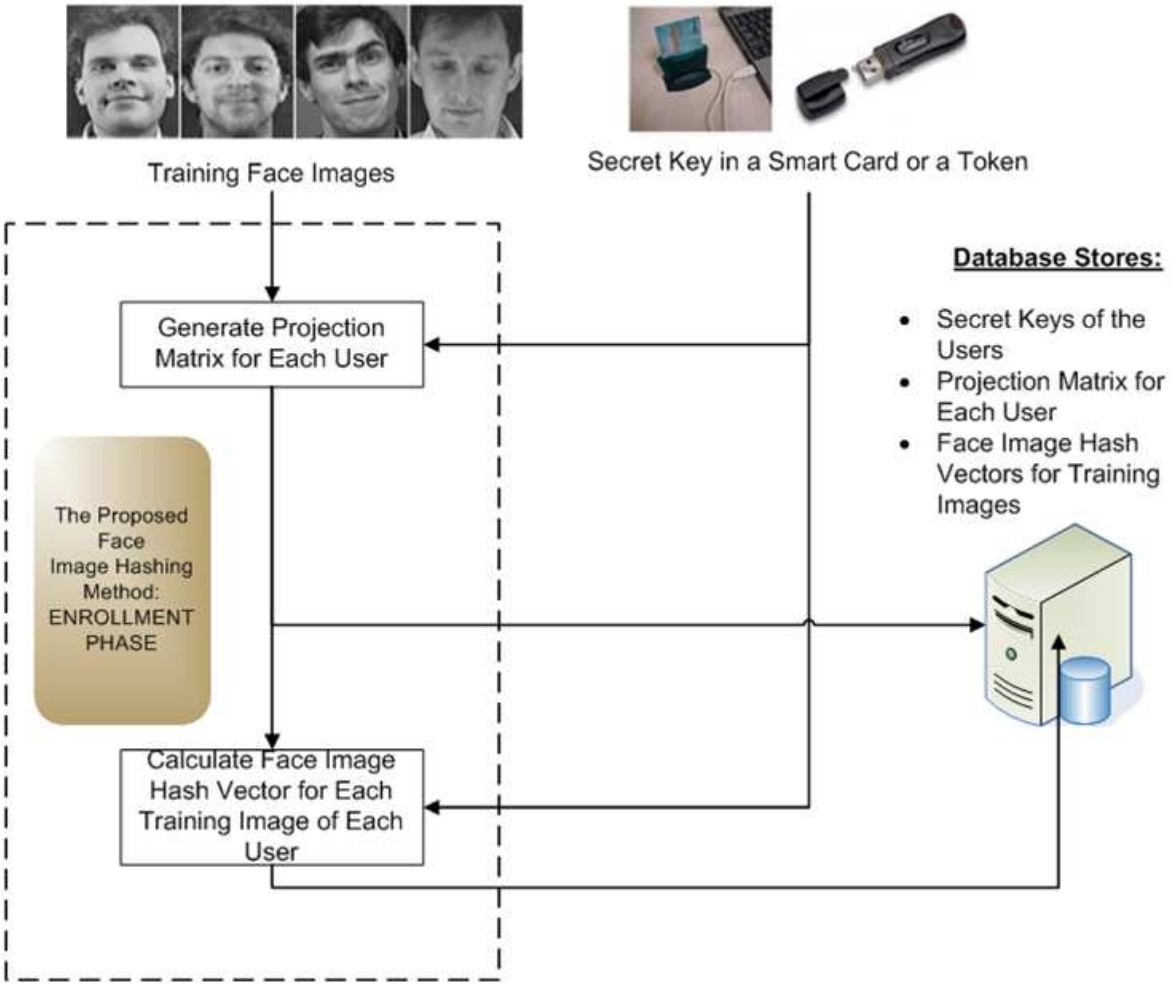

FIGURE 3.1: Illustration of enrollment stage for the proposed face image hashing method based on within-class covariance matrix

2. Optimal Linear projection,

3. Quantization.

The enrollment stage is illustrated in Figure 3.1.

3.2.1.1 Feature Extraction Phase

In the feature extraction phase, we first take training face image, $\mathbf{I}_{i,j} \in \mathbb{R}^{m \times n}$, $i = 1, \ldots, K$ and $j = 1, \ldots, L$, where K and L denote number of users and number of images per user respectively. In simulations, we use two methods in order to extract features from face image. These methods are well known methods in the literature:

1. Discrete Wavelet Transform (DWT) [89] only,

2. Discrete Wavelet Transform followed by Principle Component Analysis (PCA) [34].

Case 1: In this case, we compute the one-level Discrete Wavelet Transform (DWT) of the training face images with Haar filter. Then, we only use the coarse level, $\mathbf{X}_{i,j} \in \mathbb{R}^{(m/2)\times(n/2)}$, in order to represent each training face image. Next, we lexicographically re-order the training face images and obtain training face vectors, $\mathbf{x}_{i,j} \in \mathbb{R}^{P\times 1}$ where $P = (m/2) \times (n/2)$.

Case 2: In this case, we compute the one-level Discrete Wavelet Transform (DWT) of the training face images with Haar filter. Then, we only use the coarse level, $\mathbf{X}_{i,j} \in \mathbb{R}^{(m/2)\times(n/2)}$, in order to represent each training face image. Then, we apply principal component analysis in order to reduce the dimension of the coarse level of discrete wavelet transform $\mathbf{X}_{i,j}$ and obtain training face vectors, $\mathbf{x}_{i,j} \in \mathbb{R}^{P\times 1}$ where P denotes the number of principal components.

3.2.1.2 Optimal Linear Projection Phase

In the literature, Mihcak *et al.* [88] have defined the original data at transmitter side with X and its noisy version as noisy data with Y. In this scenario, they assumed that channel noise (additive colored Gaussian noise) is independent from the original data, X, and have a zero-mean Gaussian distribution with covariance matrix Σ_e. In their work, they have derived the set of optimal linear transforms for dimension reduction minimizing the probability of error in communication. They have also stated that this optimal linear transformation can be used in robust signal hashing problems.

In biometric hashing methods, a genuine user has a number of biometric data that is captured at different enrollment and test sessions. This causes the genuine user to have a set of associated biometric hash vectors. The different biometric data belonging to the same genuine user can be seen as noisy versions of the same biometric data. Thus, this noise can be modeled as channel noise between a transmitter and a receiver. In this work, we model the difference between the biometric data of a legitimate user and his average biometric data as channel noise. In our scenario, a regularized maximum likelihood estimate of the noise covariance which is shared across users is the within-class covariance matrix defined as follows:

$$\Sigma_e \triangleq \frac{1}{KL}\sum_{i=1}^{K}\sum_{j=1}^{L}\left(\mathbf{x}_{i,j}-\boldsymbol{\mu}_i\right)\left(\mathbf{x}_{i,j}-\boldsymbol{\mu}_i\right)^T + \alpha\mathbf{I}, \tag{3.1}$$

where $\alpha\mathbf{I}$ is added for regularization purposes, $\mathbf{U} \in \mathbb{R}^{P\times P}$ is the identity matrix and α is called the regularization parameter. Furthermore,

$$\boldsymbol{\mu}_i \triangleq \frac{1}{L}\sum_{j=1}^{L}\mathbf{x}_{i,j} \tag{3.2}$$

is the class centroid (the transmitted data in the communication scenario above), where $\mathbf{x}_{i,j}$ denotes the j^{th} face feature vector of the i^{th} user.

In this case, the set of projection matrices, $\mathbf{T}$, which minimizes the probability of error can be written as follows [88]:

$$\mathbf{T} \triangleq \mathbf{U}\Sigma\mathbf{V}^T, \tag{3.3}$$

where $\mathbf{T} \in \mathbb{R}^{k\times P}$ is the projection matrix, k denotes the length of the final face hash vector $\mathbf{f} \in \mathbb{R}^{k\times 1}$, $\mathbf{U} \in \mathbb{R}^{k\times k}$ is a random matrix whose elements are generated from the standard uniform distribution on the open interval $(0, 1)$ by using a Random Number Generator (RNG) with a seed derived from the password and its columns are orthonormal, $\Sigma \in \mathbb{R}^{k\times k}$is a diagonal matrix with pseudo-random positive diagonal entries which are generated from the standard uniform distribution on the open interval $(0, 1)$ by using the RNG with a seed derived from the password, $\mathbf{V}^T \in \mathbb{R}^{k\times P}$ is defined as follows:

$$\mathbf{V} \triangleq \mathbf{RH}, \tag{3.4}$$

where $\mathbf{H} \in \mathbb{R}^{k\times k}$ is a pseudo-random matrix whose elements are generated from the standard uniform distribution on the open interval $(0, 1)$ by using a RNG with a seed derived from the password and its columns are orthonormal, $\mathbf{R}$ is a column matrix containing k eigenvectors, which have the k smallest eigenvalues of Σ_e which has the following eigen-decomposition:

$$\Sigma_e \triangleq \mathbf{GZG}^T, \tag{3.5}$$

where $\mathbf{G}$ is a column matrix containing the eigenvectors of Σ_e, $\mathbf{Z}$ is a diagonal matrix whose diagonal elements are the corresponding eigen-values sorted from the highest magnitude to the lowest. The matrix $\mathbf{R}$ consists of the k rightmost columns of the matrix $\mathbf{G}$, the eigenvectors corresponding to the lowest magnitude eigen-values.

Thus, we can define the projection matrix $\mathbf{T}$ as follows:

$$\mathbf{T} \triangleq \mathbf{U}\Sigma\mathbf{V}^T \triangleq \mathbf{U}\Sigma\mathbf{H}^T\mathbf{R}^T. \tag{3.6}$$

Consequently, we will use this projection matrix which is partially generated in a random fashion in order to generate the final hash vector. Intuitively, the matrix $\mathbf{T}$ seeks to project the data to a lower dimensional subspace where the energy of the projected noise component is lowest.

At the next step, we project the training face images, $\mathbf{I}_{i,j} \in \mathbb{R}^{m\times n}$, onto a lower k-dimensional subspace as follows:

$$\mathbf{z}_{i,j} = \mathbf{T}\mathbf{x}_{i,j}, \tag{3.7}$$

where $\mathbf{z}_{i,j} \in \mathbb{R}^{k\times 1}$ denotes the raw face image hash vector for j^{th} face image of the i^{th} user.

3.2.1.3 Quantization Phase

At this phase, we quantize the elements of the raw face hash vector $\mathbf{z}_{i,j} \in \mathbb{R}^{k\times 1}$ in order to obtain the intermediate hash vector $\mathbf{q}_{i,j} \in \mathbb{R}^{k\times 1}$ of the training face image $\mathbf{I}_{i,j}$. We employ min-max method for performing the quantization [90]. This function maps the elements of the raw hash vector to the interval $[0, 1]$. This new step brings extra unpredictability to our algorithm. The quantization function is as follows;

$$\boldsymbol{q}_{i,j}(\ell) = \frac{\boldsymbol{q}_{i,j}(\ell) - \min(\boldsymbol{q}_{i,j})}{\max(\boldsymbol{q}_{i,j}) - \min(\boldsymbol{q}_{i,j})}, \tag{3.8}$$

where $\ell = 1, \ldots, k$ and $\boldsymbol{q}_{i,j} \in [0, 1]^{k\times 1}$, min(.) function computes the minimum value in its input vector whereas max(.) function computes the maximum value in its input vector. Finally, elements of the intermediate hash vector $\boldsymbol{q}_{i,j}$ is rounded to 0 or 1 according to a threshold, which is the mean value of the $\boldsymbol{q}_{i,j}$ vector as follows:

$$\boldsymbol{f}_{i,j}(\ell) = \begin{cases} 1 & \text{if} \quad \boldsymbol{q}_{i,j}(\ell) \geq \boldsymbol{\mu}, \\ 0 & \text{Otherwise}, \end{cases} \tag{3.9}$$

where $\boldsymbol{\mu}$ denotes the mean value of the elements of the vector $\boldsymbol{q}_{i,j}$ and $\boldsymbol{f}_{i,j} \in \{0,1\}^{k\times 1}$ is the binary hash vector for the training image $\mathbf{I}_{i,j}$. Finally, we store all training face image hash vectors in the database for verification purposes during the test stage.

3.2.2 Authentication Stage

At this stage, a claimer aims to gain access to the system via her biohash. For extracting binary hash vectors, the same phases as in the enrollment stage are used in the test stage as well. The authentication stage is illustrated in Figure 3.2.

3.2.2.1 Feature Extraction Phase

In the authentication stage, we first take a test face image, $\tilde{\mathbf{I}}_{i,j} \in \mathbb{R}^{m\times n}$ (or the system captures a face image of a claimer). Then, we use the same biometric feature extraction method used in the enrollment stage. Recall that we have two cases in the enrollment stage. In case 1, we perform DWT whereas we perform DWT followed by PCA in case 2 as addressed in the enrollment stage section. Then, we apply the same procedures as in the enrollment stage in order to compute the face image hash vector of the test image.

3.2.2.2 Optimal Linear Projection and Quantization Phases

In this part, we first reduce the dimensionality of the test face vector, $\tilde{\boldsymbol{x}}$, by multiplying it with the projection matrix, $\mathbf{T}$, as follows:

$$\tilde{\mathbf{z}} = \mathbf{T}\tilde{\mathbf{x}}, \tag{3.10}$$

where $\tilde{\boldsymbol{x}} \in \mathbb{R}^{k\times 1}$ denotes the raw face hash vector. Note that the random parts of $\mathbf{T}$ are randomly generated using the secret key provided by the user. At the next step, we process the elements of the raw face hash vector $\tilde{\boldsymbol{z}}$ in order to obtain the intermediate hash vector $\tilde{\mathbf{q}} \in [0,1]^{k\times 1}$ of the test image $\tilde{\mathbf{I}}$ as we have discussed in the enrollment section. By making threshold operation to the intermediate hash vector, we obtain the binary hash vector $\boldsymbol{f} \in [0,1]^{k\times 1}$ for the test image . For the verification decision, we calculate the Hamming distance between the hash vector of test face image and the hash vectors of the training images belonging to the claimer and stored in

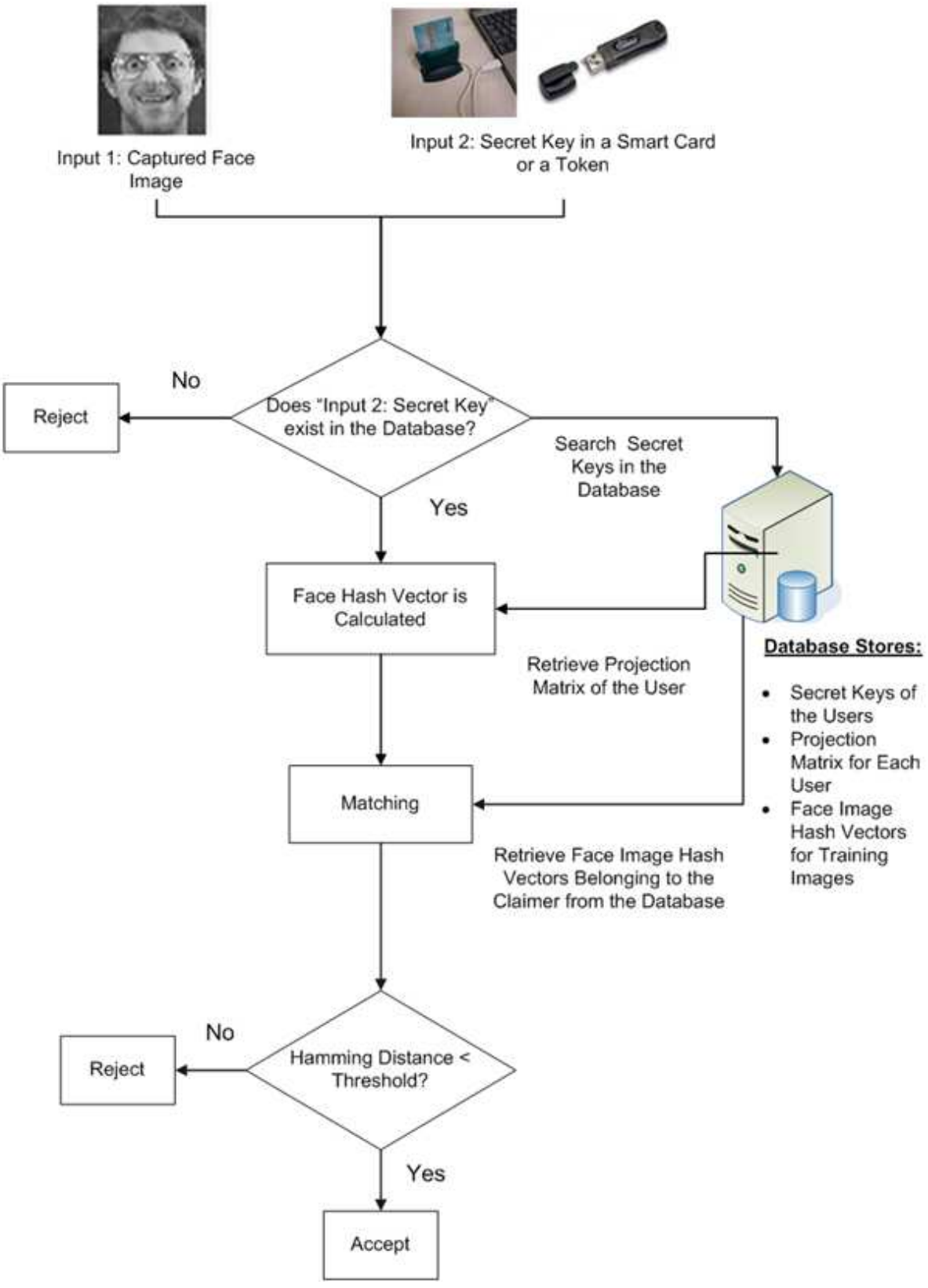

FIGURE 3.2: Illustration of enrollment stage for the proposed face image hashing method based on within-class covariance matrix

the database. If Hamming distance is below the pre-determined distance threshold, the claimer is accepted; otherwise, the user is rejected as show in Figure 3.2.

3.3 Simulation Results

In this section, we report our experimental results of the proposed face image hashing method using the optimal linear transform. First, we demonstrate the performance of the proposed method on Cambridge university AT&T face database [91]. There are 400 different face images from 10 different images of each of 40 distinct subjects. The images were taken at different times, varying the lighting, facial expressions and facial details. The size of the face images are $m = 112$ and $n = 92$. Some sample images from AT&T face database are shown in Figure 3.3. We have divided the database into two sets: training set and test set. The training set consists of 5 images per user and the test set includes also 5 images per user. Secondly, we test the performance of the proposed method on face images extracted from the Multi Modal Verification for Teleservices and Security applications (M2VTS) face database [92, 93]. The face images in this database are taken from the image frames of the video sequences as described in [93]. That database consists of face images with various expressions, illumination conditions, angles, age, sex, and glasses. The size of the face images are $m = 48$ and $n = 64$. There are 1480 face images, which consist of 40 different face poses for 37 different people in the database. We have divided the database into two sets: training set and test set. The training set consists of 20 images per user and the test set includes also 20 images per user.

We employ Hamming distance to measure the distance between the hash values of the face images. By looking at all possible pairs, we test the performance of the system. In simulations, we also set the regularization parameter α to 3 in Equation 3.1.

In the simulations with the AT&T database: for feature extraction method in case 1, we set $L = 2576$ due to face image dimensions; for feature extraction in case 2, we set L=1024 since we perform PCA after DWT. In the simulations with the M2VTS database: for feature extraction method in case 1, we set $L = 768$ due to face image dimensions; for feature extraction in case 2, we again set $L = 768$ since we perform full PCA after DWT.

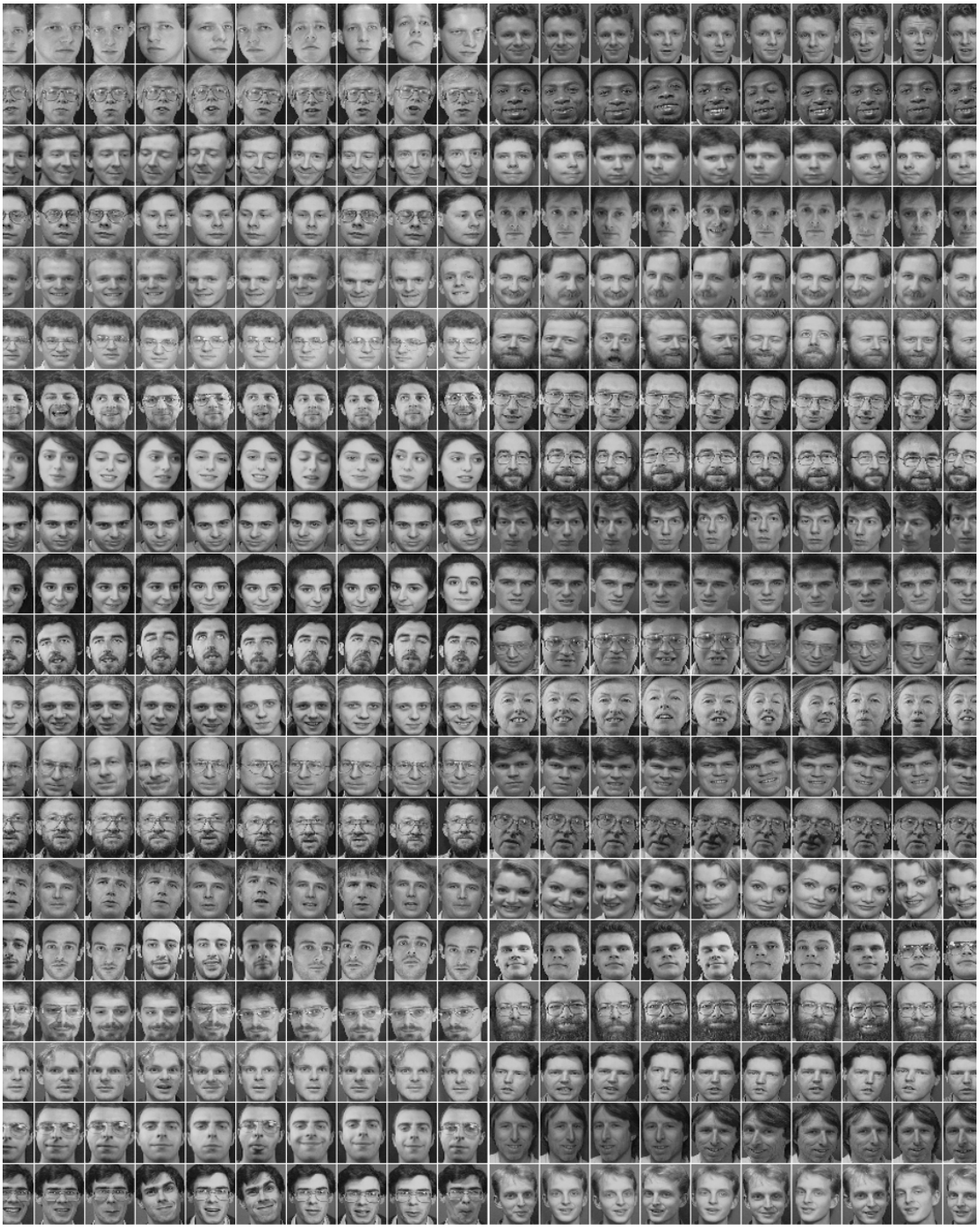

FIGURE 3.3: A preview image of the AT&T face database.

3.3.1 Experiments

In this sub-section, we study the performance of the proposed biometric hashing method using standard verification metrics and compare it with the random projection based biometric hashing methods proposed in the literature [2, 3].

Before implementing biometric hashing methods, we do not apply any pre-processing to the face images in AT&T and M2VTS face image databases (i.e. alignment, illumination normalization, rotation etc.).

Key-unknown Scenario: We assume that unauthorized users (attacker) have neither the secret key (password or PIN) nor the biometric data (face image).

Key-stolen Scenario: We assume that unauthorized users (the attacker) steal the secret key (password or PIN) and they can compute the person specific projection matrix T; however, they do not have the biometric data of the claimed genuine person, so they use other biometric data (face image of other users in the database)

In this scenario, we simulate the performance of the biometric hashing method in case an attacker takes hold of the secret key of a legitimate user. Thus, we evaluate the actual performance of the biometric hashing methods in this scenario since the secret key may be lost at any time. Besides, the performance of the biometric hashing methods should not depend on the secrecy of the user dependent key [17]. Otherwise, if the key can be kept as a secret, we would not need biometrics at all.

The simulation results of the key-unknown scenario are given in Table 3.1-Table 3.4 in terms of equal error rate (EER). The error rates are mostly zero since the randomization of the projections enables perfect separation of genuine and impostor users. In other words, since the user-defined secret key is not acquired by the attacker, we achieve perfect separation between genuine and imposter pairs. The simulation results of the key-stolen scenario are given in Table 3.5-Table 3.8 in terms of EER. It is clearly seen that the introduced method is superior to the method in [2, 3] since we make use of the noise subspace properties in an intelligent manner.

3.4 Chapter Summary

In this chapter, we have proposed a new face image hashing method based on optimal linear transformation defined for a noisy communication channel between transmitter and receiver [88]. We model the variations between the face images belonging to the same user as channel noise. In our case, the maximum likelihood estimate of the noise covariance is the within-class covariance matrix. We evaluate the performance of the biometric verification system based on the proposed face image hashing method on AT&T and M2VTS face databases with two different use-case scenarios. The simulation results show that the proposed biometric hashing method has much better performance in comparison to the random projection based biometric hashing methods in the literature [2, 3]. Furthermore, in general, performance of the proposed method increases with the increasing length of the biometric hash vector. Finally, the proposed method can also be applied to other biometrics such as fingerprint and iris.

TABLE 3.1: The EERs of the proposed face image hashing method and Ngo *et al.*s method [2, 3] for key-unknown scenario with feature extraction method in case 1 (DWT only) and with AT&T face database

Length	EER (%) of the Proposed Method	EER (%) of EER of Ngo *et al.*s Method [2, 3]	Scenario	Method	Database
64 bit	% 0.00	% 0.00	Key-unknown	Case 1 (DWT only)	AT&T
128 bit	% 0.00	% 0.00	Key-unknown	Case 1 (DWT only)	AT&T
256 bit	% 0.00	% 0.00	Key-unknown	Case 1 (DWT only)	AT&T
512 bit	% 0.00	% 0.00	Key-unknown	Case 1 (DWT only)	AT&T

TABLE 3.2: The EERs of the proposed face image hashing method and Ngo *et al.*s method [2, 3] for key-unknown scenario with feature extraction method in case 2 (DWT plus PCA) and with AT&T face database

Length	EER (%) of the Proposed Method	EER (%) of EER of Ngo *et al.*s Method [2, 3]	Scenario	Method	Database
64 bit	% 0.05	% 0.04	Key-unknown	Case 2 (DWT plus PCA)	AT&T
128 bit	% 0.02	% 0.00	Key-unknown	Case 2 (DWT plus PCA)	AT&T
256 bit	% 0.00	% 0.00	Key-unknown	Case 2 (DWT plus PCA)	AT&T
512 bit	% 0.00	% 0.00	Key-unknown	Case 2 (DWT plus PCA)	AT&T

TABLE 3.3: The EERs of the proposed face image hashing method and Ngo *et al.*s method [2, 3] for key-unknown scenario with feature extraction method in case 1 (DWT only) and with M2VTS face database

Length	EER (%) of the Proposed Method	EER (%) of EER of Ngo *et al.*s Method [2, 3]	Scenario	Method	Database
64 bit	% 0.00	% 0.00	Key-unknown	Case 1 (DWT only)	M2VTS
128 bit	% 0.00	% 0.00	Key-unknown	Case 1 (DWT only)	M2VTS
256 bit	% 0.00	% 0.00	Key-unknown	Case 1 (DWT only)	M2VTS
512 bit	% 0.00	% 0.00	Key-unknown	Case 1 (DWT only)	M2VTS

TABLE 3.4: The EERs of the proposed face image hashing method and Ngo *et al.*s method [2, 3] for key-unknown scenario with feature extraction method in case 2 (DWT plus PCA) and with M2VTS face database

Length	EER (%) of the Proposed Method	EER (%) of EER of Teoh *et al.*s Method [2, 3]	Scenario	Method	Database
64 bit	% 15.36	% 17.88	Key-unknown	Case 2 (DWT plus PCA)	M2VTS
128 bit	% 09.03	% 13.18	Key-unknown	Case 2 (DWT plus PCA)	M2VTS
256 bit	% 06.39	% 10.22	Key-unknown	Case 2 (DWT plus PCA)	M2VTS
512 bit	% 04.43	% 08.64	Key-unknown	Case 2 (DWT plus PCA)	M2VTS

TABLE 3.5: The EERs of the proposed face image hashing method and Ngo *et al.*s method [2, 3] for key-stolen scenario with feature extraction method in case 1 (DWT only) and with AT&T face database

Length	EER (%) of the Proposed Method	EER (%) of EER of Ngo *et al.*s Method [2, 3]	Scenario	Method	Database
64 bit	% 5.91	% 32.05	Key-stolen	Case 1 (DWT only)	AT&T
128 bit	% 12.86	% 28.15	Key-stolen	Case 1 (DWT only)	AT&T
256 bit	% 12.58	% 23.47	Key-stolen	Case 1 (DWT only)	AT&T
512 bit	% 18.23	% 20.84	Key-stolen	Case 1 (DWT only)	AT&T

TABLE 3.6: The EERs of the proposed face image hashing method and Ngo *et al.*s method [2, 3] for key-stolen scenario with feature extraction method in case 2 (DWT plus PCA) and with AT&T face database

Length	EER (%) of the Proposed Method	EER (%) of EER of Ngo *et al.*s Method [2, 3]	Scenario	Method	Database
64 bit	% 13.98	% 32.05	Key-stolen	Case 2 (DWT plus PCA)	AT&T
128 bit	% 13.45	% 28.15	Key-stolen	Case 2 (DWT plus PCA)	AT&T
256 bit	% 13.29	% 23.47	Key-stolen	Case 2 (DWT plus PCA)	AT&T
512 bit	% 15.85	% 20.84	Key-stolen	Case 2 (DWT plus PCA)	AT&T

TABLE 3.7: The EERs of the proposed face image hashing method and Ngo *et al.*s method [2, 3] for key-stolen scenario with feature extraction method in case 1 (DWT only) and with M2VTS face database

Length	EER (%) of the Proposed Method	EER (%) of EER of Ngo *et al.*s Method [2, 3]	Scenario	Method	Database
64 bit	% 13.52	% 36.90	Key-stolen	Case 1 (DWT only)	M2VTS
128 bit	% 27.86	% 35.15	Key-stolen	Case 1 (DWT only)	M2VTS
256 bit	% 23.43	% 25.93	Key-stolen	Case 1 (DWT only)	M2VTS
512 bit	% 37.15	% 25.79	Key-stolen	Case 1 (DWT only)	M2VTS

TABLE 3.8: The EERs of the proposed face image hashing method and Ngo *et al.*s method [2, 3] for key-stolen scenario with feature extraction method in case 2 (DWT plus PCA) and with M2VTS face database

Length	EER (%) of the Proposed Method	EER (%) of EER of Ngo *et al.*s Method [2, 3]	Scenario	Method	Database
64 bit	% 17.75	% 21.14	Key-stolen	Case 2 (DWT plus PCA)	M2VTS
128 bit	% 11.22	% 18.33	Key-stolen	Case 2 (DWT plus PCA)	M2VTS
256 bit	% 09.84	% 16.59	Key-stolen	Case 2 (DWT plus PCA)	M2VTS
512 bit	% 07.64	% 16.13	Key-stolen	Case 2 (DWT plus PCA)	M2VTS

Chapter 4

Discriminative Projection Selection Based Face Image Hashing

4.1 Introduction

[1] In this chapter, we propose a new face image hashing method based on a proposed technique that we call "discriminative projection selection" to reduce verification errors [26]. This technique selects the rows of the random projection matrix used in biometric hashing matrix, which is a user dependent dimension reduction matrix, by using the Fisher criterion [27]. Moreover, we employ Gaussian mixture model at the quantization step to obtain more distinct face image hash vectors for each user.

4.2 The Proposed Biometric Verification Method

In this section, we introduce a biometric verification system which employs the proposed face image hashing method based on the discriminative projection selection technique. In the proposed biometric verification method, there are two main stages:

1. Enrollment stage,
2. Authentication stage.

[1] This chapter is based on [26].

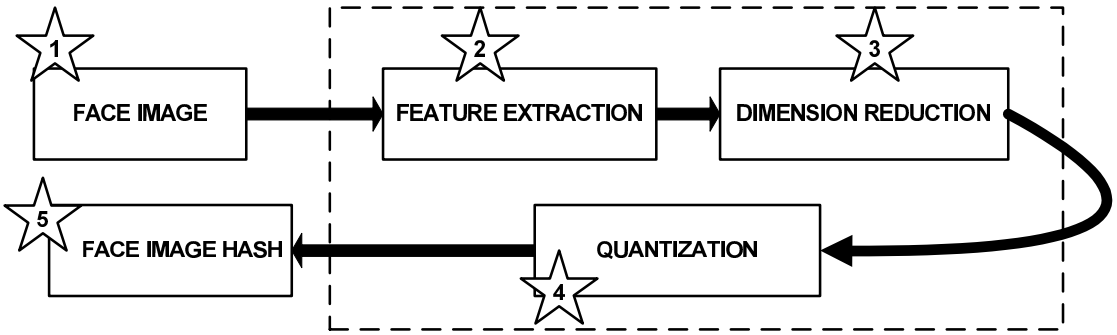

FIGURE 4.1: Basic steps of the biometric hashing methods

These stages are addressed in the below parts.

4.2.1 Enrollment Stage

There are three main steps at the enrollment stage:

1. Feature extraction,
2. Dimension reduction,
3. Quantization.

4.2.1.1 Feature Extraction

At the feature extraction phase, we use two sets of data: training set and "others" set. The training set has training face images of registered users, $\mathbf{I}_{i,j} \in \mathbb{R}^{m \times n}$, $i = 1, \ldots, K$ where K denotes number of users and $j = 1, \ldots, L$ where L denotes number of training images per user. We lexicographically re-order them and obtain training face vectors, $x_{i,j} \in \mathbb{R}^{(mn) \times 1}$. The others set contains randomly selected face images which do not belong to any registered users $\tilde{\mathbf{I}}_s \in \mathbb{R}^{m \times n}$, $s = 1, \ldots, M$ where M denotes the number of face images belonging to the others set. We again lexicographically re-order them and obtain face vectors, $\tilde{\mathbf{x}}_s \in \mathbb{R}^{(mn) \times 1}$ of the others set. We apply PCA to the face images in the training set for feature extraction.

$$\mathbf{y}_{i,j} = \mathbf{A}(\mathbf{x}_{i,j} - \boldsymbol{\mu}), \tag{4.1}$$

where $\mathbf{A} \in \mathbb{R}^{q\times(mn)}$ is the PCA matrix trained using the face images in the training set, $\boldsymbol{y}_{i,j} \in \mathbb{R}^{q\times 1}$ is the PCA coefficient vector belonging to the j^{th} training image of the i^{th} user and $\boldsymbol{\mu}$ is the mean face vector. We project the face images in the others set onto the PCA subspace as follows:

$$\tilde{\boldsymbol{y}}_s = \mathbf{A}(\tilde{\boldsymbol{x}}_s - \boldsymbol{\mu}), \tag{4.2}$$

where $\tilde{\boldsymbol{y}}_s \in \mathbb{R}^{q\times 1}$ is the PCA coefficient vector belonging to the s^{th} image of the others set. We use these PCA coefficient vectors in the discriminative projection selection technique to find the most valuable features, which maximize the distance between the face images of a user in the training set and the face images in the others set, in the lower-dimensional subspace.

4.2.1.2 Dimension Reduction

At the dimension reduction phase, we first generate a random projection (RP) matrix, $\mathbf{T}_i \in \mathbb{R}^{\ell\times q}$, for each user to reduce the dimension of her feature vector. The RP matrix elements are identically and independently distributed (*i.i.d*) and generated from a Gaussian distribution with zero mean and unit variance by using a random number generator (RNG) with a seed derived from the user's secret key. We apply the Gram-Schmidt (GS) procedure to obtain an orthonormal projection matrix $\mathbf{R}_i \in \mathbb{R}^{\ell\times q}$ to have more distinct projections. Then, we project the PCA coefficient vectors of the i^{th} user onto a lower ℓ-dimensional subspace.

$$\boldsymbol{z}_{i,j} = \mathbf{R}_i \mathbf{y}_{i,j}, \tag{4.3}$$

where $\boldsymbol{z}_{i,j} \in \mathbb{R}^{\ell\times 1}$ is the intermediate face image hash vector belonging to the j^{th} training image of the i^{th} user.

To determine the competing hash vectors, we also project the others set using the RP matrix as follows:

$$\tilde{\mathbf{z}}_s = \mathbf{R}_i \tilde{\mathbf{y}}_s, \tag{4.4}$$

where $\tilde{\mathbf{z}}_s \in \mathbb{R}^{\ell\times 1}$ is the intermediate face image hash vector of the s^{th} face image of the others set and $\mathbf{R}_i$ is the orthonormal random projection matrix of the i^{th} user.

The proposed discriminative projection selection technique selects the rows of the matrix $\mathbf{R}_i$ using the Fisher criterion [27] and creates the discriminative random projections. Thus, we aim to increase discriminability due to mapping the PCA coefficient vectors into a more discriminant subspace. Fisher criterion is a feature selection method and in this case the features are obtained after the random projection of a PCA coefficient vector $\boldsymbol{y}$, namely:

$$z(k) = \boldsymbol{r}_{i,k}^T \boldsymbol{y}, \tag{4.5}$$

for $k = 1, \ldots, \ell$, where $z(k)$ is a scalar value and $\boldsymbol{r}_{i,k}^T \in \mathbb{R}^{1\times q}$ denotes the k^{th} row of $\mathbf{R}_i$. The k^{th} feature for the i^{th} user's j^{th} image is $z_{i,j}(k)$ and the same feature for the s^{th} element of the others set is $\tilde{z}_s(k)$ which are the k^{th} elements of the corresponding vectors defined in Equations (4.3) and (4.4). The features are already uncorrelated due to the GS procedure which ensures $\boldsymbol{r}_{i,k}^T \boldsymbol{r}_{i,m} = 0$ for $k \neq m$.

We define

$$\boldsymbol{e}_i^k \triangleq \left[z_{i,1}(k), \ldots, z_{i,L}(k)\right], \tag{4.6}$$

which is a collection of the k^{th} dimension (or bit position) coefficients of the intermediate hash vectors belonging to the i^{th} user for each $k = 1, \ldots, \ell$.

First, we compute the sample mean value, $\hat{\mu}_i^{1,k}$, of each $\boldsymbol{e}_i^k$ vector for each bit position $k = 1, \ldots, \ell$ as follows:

$$\hat{\mu}_i^{1,k} = \frac{1}{L} \sum_{j=1}^{L} e_i^k(j), \tag{4.7}$$

where $e_i^k(j)$ is the j^{th} element of the vector $\boldsymbol{e}_i^k$ which is defined in Equation (4.6).

Then, we compute the sample standard deviation, $\hat{\sigma}_i^{1,k}$, of each e_i^k vector for each bit position $k = 1, \ldots, \ell$ as follows:

$$\hat{\sigma}_i^{1,k} = \left(\frac{1}{L} \sum_{j=1}^{L} \left(e_i^k(j) - \hat{\mu}_i^{1,k} \right)^2 \right)^{\frac{1}{2}}. \tag{4.8}$$

Similarly, we collect together the k^{th} dimension values of the intermediate hash vectors of the "others" data set as follows:

$$\boldsymbol{q}^k \triangleq [\tilde{z}_1(k), \ldots, \tilde{z}_M(k)], \tag{4.9}$$

for all $k = 1, \ldots, \ell$.

First, we compute the sample mean value, $\hat{\mu}_i^{2,k}$, of each $\boldsymbol{q}^k$ for each bit position $k = 1, \ldots, \ell$ as follows:

$$\hat{\mu}_i^{2,k} = \frac{1}{M} \sum_{s=1}^{M} q^k(s), \tag{4.10}$$

where $q^k(s)$ is the s^{th} element of the vector $\boldsymbol{q}^k$ which is defined in Equation (4.9).

Next, we compute the sample standard deviation, $\hat{\sigma}_i^{2,k}$, of each $\boldsymbol{q}^k$ for each bit position $k = 1, \ldots, \ell$ as follows:

$$\hat{\sigma}_i^{2,k} = \left(\frac{1}{M} \sum_{s=1}^{M} \left(\boldsymbol{q}^k(s) - \hat{\mu}^{2,k} \right)^2 \right)^{\frac{1}{2}}. \tag{4.11}$$

By applying the Fisher criterion, we try to select the rows that have higher contrast between genuine user's data and the others set. In other words, we aim to reduce the distance between the genuine user's different face image hash vectors while at the same time we aim to maximize the distance between the i^{th} user's data and the others set. We compute the Fisher score for each row of $\mathbf{R}_i$ as follows:

$$\eta_i(k) = \frac{\left| \hat{\mu}_i^{1,k} - \hat{\mu}_i^{2,k} \right|^2}{\left(\hat{\sigma}_i^{1,k} \right)^2 + \left(\hat{\sigma}_i^{2,k} \right)^2}. \tag{4.12}$$

We use these Fisher scores obtained for each dimension k to rank the rows of the random projection matrix $\boldsymbol{R}_i$. We define $\boldsymbol{r}_{i,k}^T$ to be the k^{th} row of $\mathbf{R}_i$. That is

$$\mathbf{R}_i = \begin{pmatrix} \boldsymbol{r}_{i,1}^T \\ \vdots \\ \boldsymbol{r}_{i,\ell}^T \end{pmatrix}. \tag{4.13}$$

We choose top ranking rows from the random projection matrix. Let $\boldsymbol{c}_i$ be the index vector which ranks the rows of the matrix in a descending manner from 1 to w where w is the number of desired rows. That is, $c_i(1)$ is the row index of $\mathbf{R}_i$ that has the highest Fisher score, $c_i(2)$ is

the index with the second highest score and so on. Thus, we obtain the discriminative random projection matrix $\hat{\mathbf{R}}_i \in \mathbb{R}^{w \times q}$ and the index vector, $\boldsymbol{c}_i$, which contains the indices of top w rows for each user. We define

$$\hat{\mathbf{R}}_i = \begin{pmatrix} \hat{\boldsymbol{r}}_{i,1}^T \\ \vdots \\ \hat{\boldsymbol{r}}_{i,w}^T \end{pmatrix}, \tag{4.14}$$

where $\hat{\boldsymbol{r}}_{i,p} = \boldsymbol{r}_{i,c_i(p)}$ for $p = 1, \ldots, w$. We only store the index vector, $\boldsymbol{c}_i$, for the i^{th} user in the database for verification at the test stage.

Next, we project the PCA coefficients, which belong to the training face images of the i^{th} user, onto a lower w-dimensional subspace by using the calculated $\hat{\mathbf{R}}_i$ as follows:

$$\boldsymbol{f}_{i,j} = \hat{\mathbf{R}}_i y_{i,j}, \tag{4.15}$$

where $\boldsymbol{f}_{i,j} \in \mathbb{R}^{w \times 1}$ is the raw face image hash vector belonging to the j^{th} training image of the i^{th} user.

Ngo *et.al* [2, 3] uses FLD as a feature extraction method which is applied before the random projection step in the algorithm. Their FLD transform is not user specific and aims to discriminate face images belonging to different users in the database. In our case, we employ the Fisher criterion for projection selection for biometric verification. Therefore, in our case, the projection selection is user-specific and aims to discriminate the claimed user's biometric hash vector from all other possible ones that may come from other face images. In our case, other face images may even be from outside the database which is more realistic in a real scenario. Our others set is chosen from another database for this purpose. The projection selection is done after the random projection step. FLD can reduce dimension at most to $K - 1$ dimensions, where K is the number of users in the database, due to the maximal rank of the between class covariance matrix. However, in our method, we do not have such a limitation since the selection is done by ranking the Fisher criteria obtained from each projection. In summary, there are fundamental differences between using FLD as a dimension reducing feature extraction method and using the Fisher criteria for selection of random projections that best separate the claimed identity from all others.

4.2.2 Quantization

In this subsection, we discuss the quantization methods used in this work. We employ two different quantization methods: (1) Binary quantization (BQ) [2, 3] and (2) The proposed Gaussian mixture model (GMM) based quantization method. In our simulations, we employ these quantization methods separately to show the performance of the system.

4.2.2.1 Binary Quantization Method with a Fixed Threshold

This technique is employed in Ngo *et al.*'s method [2, 3]. The raw face image hash vector $\boldsymbol{f}_{i,j}$ elements are binarized with respect to a fixed threshold as follows:

$$\lambda_{i,j}(k) = \begin{cases} 1 & \text{if } \boldsymbol{f}_{i,j}(k) \geq \tilde{\mu}, \\ 0 & \text{Otherwise,} \end{cases} \tag{4.16}$$

where the threshold $\tilde{\mu}$ is chosen as the sample mean value of elements of the vector $\boldsymbol{f}_{i,j}$ and $k = 1, \ldots, w$. The computed reference face image hash vectors $\boldsymbol{\lambda}_{i,j} \in \mathbb{R}^{w \times 1}$ are stored in the database.

4.2.2.2 The Proposed GMM Based Quantization Method

To the best of our knowledge, there is no face image hashing method employing GMM in the quantization step. GMM is one of the most widely used data clustering methods in the literature [94]. Let us assume that we have a set of numbers which are obtained by collecting the k^{th} elements of raw face image hash vectors and we want to binarize the element of this set. Since our aim is to make binarization, we fit two Gaussian distributions to the histogram of the k^{th} elements of the raw face image hash vectors by using the GMM. Then, we choose the average of the mean values of these two Gaussian distributions as a meaningful threshold for partition of these two distributions. We repeat it for each bit location $k = 1, \ldots, w$ separately. In other words, we employ bimodal GMM to find a threshold for each bit position for binarization. Let $f_{i,j}(k)$ denote the k^{th} bit of $\boldsymbol{f}_{i,j} \in \mathbb{R}^{w \times 1}$, we define the vector $\boldsymbol{d}_k$ as the collection of all k^{th} dimension values of the raw image hashes in the database.

$$\boldsymbol{d}_k \triangleq \left[\boldsymbol{f}_{i,j}(k) : i = 1, \ldots, K, j = 1, \ldots, L\right] \in \mathbb{R}^{r \times 1}, \tag{4.17}$$

where $k = 1, \ldots, w$, $r = K \times L$, K is the number of users and L is the number of training images per user. Assume that the elements of the vector $\boldsymbol{d}_k$ are observations of a single random variable d.

$$p\left(d \mid \Psi^k\right) = \sum_{s=1}^{S} \alpha_s^k p\left(d \mid \theta_s^k\right), \tag{4.18}$$

where α_s^k is a mixture weight, $\sum_{s=1}^{S} \alpha_s^k = 1$ where $S = 2$ due to binarization, $\Psi^k = \left\{\alpha_1^k, \alpha_2^k, \theta_1^k, \theta_2^k\right\}$ and $p\left(d \mid \Psi^k\right)$ is a one-dimensional Gaussian density with its own parameters $\theta_s^k = \left\{\mu_s^k, \sigma_s^k\right\}$ as follows:

$$p\left(d \mid \theta_s^k\right) = \frac{1}{\sigma_s^k \sqrt{2\pi}} e^{-\left(d-\mu_s^k\right)^2 / \left(2\left(\sigma_s^k\right)^2\right)}, \tag{4.19}$$

where μ_s^k and σ_s^k denote the mean and the standard deviation for the s^{th} component of the GMM respectively for $s = \{1, 2\}$. We find a threshold for each bit position as follows:

$$T_k = \frac{\mu_1^k + \mu_2^k}{2}, \tag{4.20}$$

where T_k denotes the threshold for the k^{th} bit position of the raw face image hash vector $\boldsymbol{f}_{i,j}$, $\forall i, j$. Note that, the GMM is trained using the whole training set for each bit position. Thus, the GMM parameters are not user dependent. Finally, the elements of $\boldsymbol{f}_{i,j}$ are binarized with respect to the system-level thresholds as follows:

$$\lambda_{i,j}(k) = \begin{cases} 1 & \text{if } \boldsymbol{f}_{i,j}(k) \geq T_k, \\ 0 & \text{Otherwise,} \end{cases} \tag{4.21}$$

where $k = 1, \ldots, w$. The computed reference face image hash vectors $\boldsymbol{\lambda}_{i,j} \in \mathbb{R}^{wx1}$ are stored in the database.

4.2.3 Authentication Stage

At the authentication stage, a claimer claims that she is the i^{th} user and sends her face image and her secret key to the system. The system computes her test face image hash vector by using her face image, her secret key (to generate a RP matrix) and the index vector, $\boldsymbol{c}_i$, which belongs to the i^{th} user. Recall that index vectors and the reference face image hash vectors of the registered

users are stored in the database; however, the secret keys are not stored in the database. Then, the Hamming distance [95] is computed between the test face image hash vector and the reference face image hash vectors which belong to the i^{th} user and were generated at the enrollment stage. If it is below the pre-determined distance threshold, the claimer is accepted; otherwise, the claimer is rejected.

We simulate two scenarios in our experiments. These scenarios are described in detail below.

1. ***Key-unknown Scenario:*** In this scenario, an unauthorized impostor has neither the secret key nor the face image template belonging to the genuine user. Note that the index vectors of the users are stored in the database. Therefore, whenever a claimer claims that she is the i^{th} user and sends her face image and a secret key to the system, the system computes a test face image hash vector by using the data sent by the claimer and the index vector, $\boldsymbol{c}_i$, which belongs to the i^{th} user.

2. ***Key-stolen Scenario:*** In this scenario, an unauthorized impostor acquires the secret key of the i^{th} genuine user but does not have the claimed person's face image. When an impostor sends her face image and the secret key of the i^{th} user to the system, the system computes a test face image hash vector by using the data sent by the impostor and the index vector $\boldsymbol{c}_i$ that belongs to the i^{th} user which is stored in the database.

4.3 Simulation Results

In this section, we discuss our experimental results. We test the performance of the proposed method on AT&T [91], AR [96] and Sheffield (previously UMIST) face databases [97]. The AT&T database has 400 different face images corresponding to 40 distinct people. The AR database has 3120 face images belonging to 120 different people's faces with different facial expressions, illumination conditions, and occlusions (sun glasses and scarf). Some sample images from the AR face database are shown in Figure 4.2. The Sheffield database has 564 different face images belonging to 20 different people. Some sample images from the Sheffield face database are shown in Figure 4.3. Besides, we randomly select 104 face images from Carnegie Mellon University database [98] and create the others set. Some sample images from the CMU face database are shown in Figure 4.4.

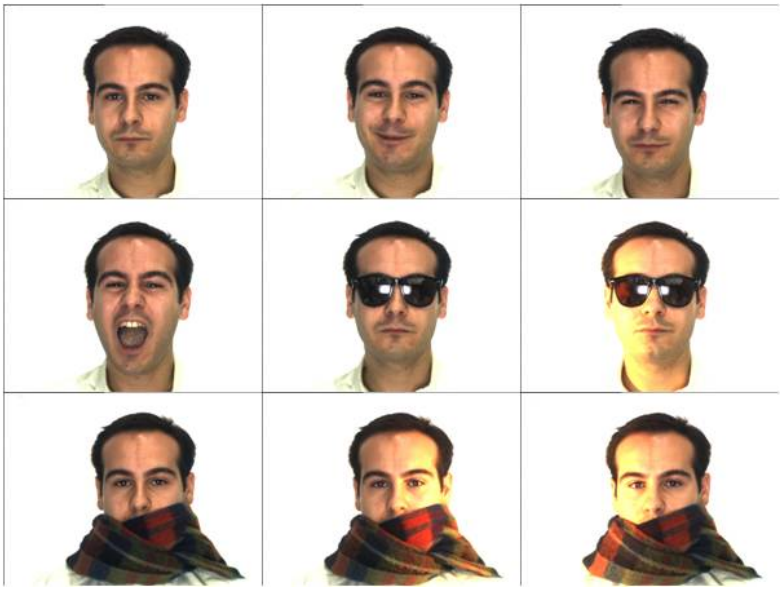

FIGURE 4.2: A preview image of the AR face database.

FIGURE 4.3: A preview image of the Sheffield face database.

FIGURE 4.4: A preview image of the CMU face database.

We compare the performance of the proposed method to Ngo *et al.*'s PCA+RP and FLD+RP methods that were introduced in [2, 3] as shown in Table 4.2. We automatically select face images for training and test sets and evaluate the performance of the proposed method and Ngo *et al.*'s methods [2, 3]. We use 1024-length PCA coefficient vectors for the face images belonging to the training, test and others sets in the simulations. In our experiments, pre-processing techniques such as eye alignment, head region masking, lighting adjustment are not applied to the face images. In our simulations for both scenarios; for impostor tests, each face image of each user in the test set is compared against each face image of all other users in the training set. A successful impostor attempt results in a false acceptance error. For the genuine tests, each face image of each user in the test set is compared against all face images of the same user in the training set. A failed genuine attempt results in a false rejection error. The detailed information on the data sets used in the experiments are given in Table 4.1.

TABLE 4.1: Datasets and experimental set-up

Database	Number of Face Images	Train set	Test set
AR	3120 images from 120 people	The first 7 images of each user	The last 2 images of each user
AT&T	400 images from 40 people	The first 5 images of each user	The remaining 5 images of each user
Sheffield	564 images from 20 people	The first 8 images of each user	The following 8 images of each user

The proposed method has better performance in terms of equal error rate (EER) in comparison to Ngo *et al.*'s methods [2, 3] whereas Ngo *et al.*'s PCA+RP and FLD+RP methods have comparable performances with each other as shown in Table 4.2. As the length of face image hash vector decreases, the proposed method shows better improvement since the proposed dimension reduction matrix better preserves the pair-wise distances between feature vectors in the reduced dimension subspace in comparison with the traditional random projection matrix. The best results are usually obtained with 128 or 256 bits. Besides, we plot the detection error trade-off (DET) curves [35] for key-stolen scenario of the 256 bit face image hash length with the AT&T database in Figure 4.5.

In chapter 3, we obtain zero EERs in the key-unknown scenario whereas very high EERs in the key-stolen scenario with the optimal linear transform based biohashing method. However, we focus to reduce the EER in the key-stolen scenario in this chapter and that's why we use only PCA for feature extraction. In addition, we do not work for finding better feature extraction methods in this thesis. It is obvious that the biohashing methods can perform better with better

TABLE 4.2: EER performances of the proposed face image hashing method and Ngo *et al.*'s methods [2, 3]

Length	EER (%) of Ngo *et al.*'s method [2, 3] (PCA+RP)	EER (%) of Ngo *et al.*'s method [2, 3] (FLD+RP)	EER (%) of the proposed method with binary quantization method	EER (%) of the proposed method with GMM based quantization method	Scenario	Database
64 bit	% 12.19	% 10.46	**% 3.83**	**% 2.73**	Key-unknown	AT&T
128 bit	% 7.36	% 8.51	**% 2.23**	**% 1.57**	Key-unknown	AT&T
256 bit	% 5.81	% 5.50	**% 1.80**	**% 1.15**	Key-unknown	AT&T
512 bit	% 3.79	% 4.17	**% 2.48**	**% 2.10**	Key-unknown	AT&T
64 bit	% 16.93	% 18.13	**% 14.40**	**% 13.58**	Key-stolen	AT&T
128 bit	% 13.97	% 16.73	**% 12.01**	**% 11.14**	Key-stolen	AT&T
256 bit	% 12.76	% 14.50	**% 10.80**	**% 10.23**	Key-stolen	AT&T
512 bit	% 12.34	% 13.55	**% 10.15**	**% 9.73**	Key-stolen	AT&T
64 bit	% 23.63	% 23.34	**% 9.08**	**% 8.96**	Key-unknown	AR
128 bit	% 18.24	% 18.05	**% 8.67**	**% 8.72**	Key-unknown	AR
256 bit	% 15.82	% 13.93	**% 7.81**	**% 8.12**	Key-unknown	AR
512 bit	% 11.38	% 11.92	**% 8.33**	**% 8.57**	Key-unknown	AR
64 bit	% 28.27	% 28.51	**% 18.07**	**% 18.46**	Key-stolen	AR
128 bit	% 27.17	% 27.56	**% 18.06**	**% 18.05**	Key-stolen	AR
256 bit	% 25.50	% 26.44	**% 19.10**	**% 18.83**	Key-stolen	AR
512 bit	% 24.89	% 25.04	**% 20.95**	**% 20.41**	Key-stolen	AR
64 bit	% 17.09	% 22.00	**% 15.75**	**% 16.23**	Key-unknown	Sheffield
128 bit	% 16.38	% 19.10	**% 13.33**	**% 14.03**	Key-unknown	Sheffield
256 bit	% 15.05	% 14.93	**% 11.45**	**% 11.05**	Key-unknown	Sheffield
512 bit	% 14.97	% 14.12	**% 10.44**	**% 12.20**	Key-unknown	Sheffield
64 bit	% 21.40	% 24.50	**% 19.38**	**% 20.68**	Key-stolen	Sheffield
128 bit	% 21.92	% 24.30	**% 17.51**	**% 19.71**	Key-stolen	Sheffield
256 bit	% 22.53	% 22.02	**% 16.96**	**% 17.80**	Key-stolen	Sheffield
512 bit	% 23.47	% 22.55	**% 19.27**	**% 18.22**	Key-stolen	Sheffield

feature extraction methods. Instead, we try to improve the performance of the random projection based biohashing methods by proposing better dimension reduction methods and/or quantization methods since these phases cause verification errors due to large information loss at these phases.

Ngo *et al.* [2, 3] employ binary quantization method with a fixed threshold for all bits. This method may be suboptimal in some cases. The proposed GMM-based quantization method reduces EER most of the time in comparison to the binary quantization since it finds more meaningful threshold values for each bit position.

4.4 Chapter Summary

In this chapter, we propose a novel biohashing method for a biometric verification system. The proposed method is based on discriminative projection selection depending on Fisher criteria. Another novelty of the proposed method is to employ bimodal GMM in the quantization. The simulations show that it has better performance in comparison with the random projection based biohashing methods proposed in the literature.

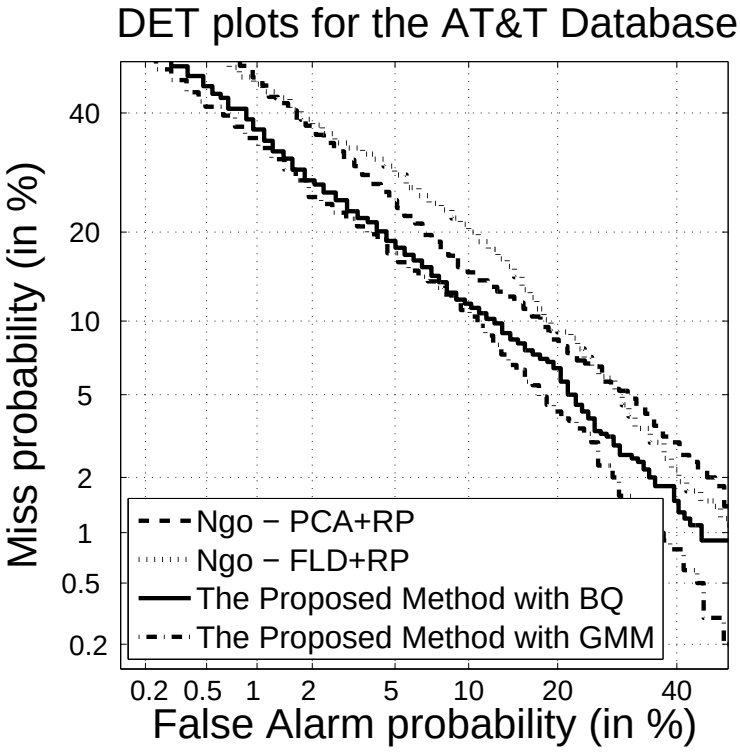

FIGURE 4.5: DET plots for the methods with 256 bit face image hash vector length for key-stolen scenario - AT&T database

Chapter 5

Error-Correcting Output Codes Guided Quantization For Biometric Hashing

5.1 Introduction

[1] In this chapter, we propose a novel biometric hashing scheme which depends on Error-Correcting Output Codes (ECOC) [28]. We improve the performance of the random projection based biometric hashing scheme by introducing a new quantization method that attempts to optimize biometric hash vectors by using the ideas from ECOC classifiers. The proposed scheme shows superior performance in comparison with Ngo *et al.*'s scheme [2] on four databases.

5.2 The Proposed Biometric Verification System

In this section, we introduce our new biometric verification system based on the proposed ECOC guided biometric hash generation method. In the proposed biometric verification method, there are two main stages:

1. Enrollment stage,

[1] This chapter is based on [28].

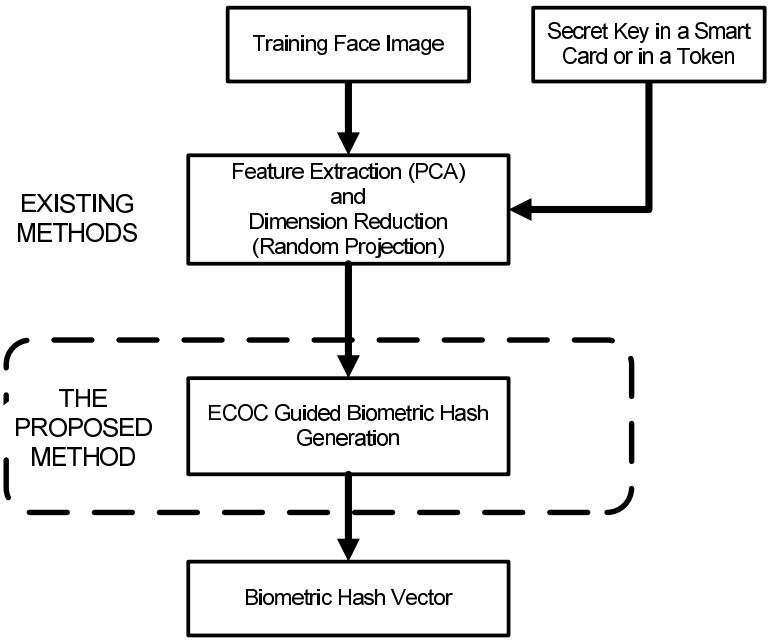

FIGURE 5.1: The basic steps of the proposed biometric hashing scheme

2. Authentication stage.

These stages are addressed in the following sections.

5.2.1 Enrollment Stage

In this part, we explain the enrollment stage which consists of three main phases:

1. Feature extraction,
2. Dimension reduction,
3. ECOC guided biometric hash generation.

5.2.1.1 Feature Extraction

At this phase, we use face images in the training set. The training set has training face images belonging to registered users, $\mathbf{I}_{i,j} \in \mathbb{R}^{m \times n}$ where $i = 1, 2, \ldots, K$ and K denotes the number of users, $j = 1, 2, \ldots, L$ and L denotes the number of training images per user. We lexicographically re-order the face images and obtain training face vectors, $x_{i,j} \in \mathbb{R}^{(mn) \times 1}$. Then, we employ

Principle Component Analysis (PCA) to the face images in the training set for feature extraction as follows:

$$\boldsymbol{y}_{i,j} = \mathbf{A}(\boldsymbol{x}_{i,j} - \boldsymbol{\mu}), \tag{5.1}$$

where $\mathbf{A} \in \mathbb{R}^{k\times(mn)}$ is the PCA matrix trained by the face images in the training set, $\boldsymbol{\mu}$ is the mean face vector and $\boldsymbol{y}_{i,j} \in \mathbb{R}^{k\times 1}$ is a vector containing PCA coefficients belonging to the j^{th} training image of the i^{th} user.

5.2.1.2 Dimension Reduction

We generate a Random Projection (RP) matrix, $\mathbf{R}_i \in \mathbb{R}^{\ell\times k}\ \forall i$, for each user to reduce the dimension of the face images in the training set. The RP matrix elements are identically and independently (*i.i.d*) generated from a Gauss distribution with zero mean and unit variance by using a Random Number Generator (RNG) with a seed derived from the user's secret key. We apply Gram-Schmidt (GS) procedure to obtain an orthonormal projection matrix $\mathbf{R}_{GS,i} \in \mathbb{R}^{\ell\times k}$ from $\mathbf{R}_i$ to have more distinct projections. We project the PCA coefficient vectors onto a lower ℓ-dimensional subspace as follows:

$$\boldsymbol{z}_{i,j} = \mathbf{R}_{GS,i}\boldsymbol{y}_{i,j}, \tag{5.2}$$

where $\boldsymbol{z}_{i,j} \in \mathbb{R}^{\ell\times 1}$ is an intermediate biometric hash vector belonging to the j^{th} training image of the i^{th} user.

5.2.1.3 ECOC Guided Biometric Hash Generation

At this phase, we first calculate a representative intermediate biometric hash vector, $\boldsymbol{E}_i$, for each user:

$$E_i(m) = \frac{1}{L}\sum_{j=1}^{L} \boldsymbol{z}_{i,j}(m), \tag{5.3}$$

where $m \in \{1, 2, \ldots, \ell\}$ and ℓ is the length of the raw biometric hash vector. Next, we map the elements of $\boldsymbol{E}_i$ to the interval $[0, 1]$ by employing min-max normalization [99] and obtain a representative raw biometric hash vector, $\boldsymbol{V}_i \in \mathbb{R}^{\ell \times 1}$, for each user as follows:

$$V_i(m) = \frac{E_i(m) - \min(\boldsymbol{E}_i)}{\max(\boldsymbol{E}_i) - \min(\boldsymbol{E}_i)}, \tag{5.4}$$

where $\boldsymbol{V}_i$ denotes a representative raw biometric hash vector of the i^{th} user, min(.) function computes minimum value of its input vector and max(.) function computes maximum value of its input vector.

Conventionally, the $\boldsymbol{V}_i$ vector is binary-quantized by thresholding to obtain the final biometric hash vector for each user. In Ngo *et al.*'s scheme [2], a different quantization threshold (t_g) for each user is obtained by computing the average value of each associated vector, that is $t_g = 1/\ell \sum_{m=1}^{\ell} V_i(m)$. Note that the threshold is the same for each bit position, therefore we can call it a *global* threshold.

In contrast, we employ bit-adaptive quantization to improve the performance of the biometric hashing scheme by generating a more diverse set of biometric hashes for authorized users. We define $\boldsymbol{C}$ as the *codeword matrix* which is formed by stacking biometric hashes of all users in its rows. The i^{th} row of $\boldsymbol{C}$ is obtained by performing quantization on $\boldsymbol{V}_i$ using a set of thresholds (one for each bit) which we aim to optimize.

$$C(i, m) = \begin{cases} 1 & \text{if} \quad \boldsymbol{V}_i(m) \geq \boldsymbol{t}(m), \\ 0 & \text{Otherwise.} \end{cases} \tag{5.5}$$

In the literature, ECOC is proposed to cope with multi-class classification problems using multiple binary classifiers [100, 101]. Here, our aim is to reduce verification errors by employing separation criteria used in ECOC classifiers to optimize the biometric hash codeword matrix $\boldsymbol{C}$ by modifying the threshold vector $\boldsymbol{t}$.

The ECOC matrices are optimized on two main criteria [102]: 1) Row separation, 2) Column separation. In the proposed method, we use row and column separation criteria described below to optimize the biometric hash vectors.

Row Separation: The Hamming distance between the biometric hash vectors, which belong to different users, should be maximized to reduce errors. The minimum Hamming distance between any pair of biometric hash vectors is called the row separation:

$$H_r(\boldsymbol{t}) = \min_{i,j,i\neq j} \sum_{m=1}^{\ell} |C(i,m) - C(j,m)|. \tag{5.6}$$

H_r is dependent on $\boldsymbol{t}$ since the thresholds determine and change the codeword matrix $\boldsymbol{C}$. An ECOC matrix with minimum Hamming distance, H_r, between any pair of biometric hash vectors will correct up to $\left\lfloor \frac{H_r-1}{2} \right\rfloor$ bit errors [10]. Thus, it is beneficial to maximize this minimum distance to obtain better biometric hash vectors.

Column Separation: Column separation is defined as the minimum Hamming distance between all the columns of the codeword matrix $\boldsymbol{C}$. The aim in ECOC matrix design is to maximize the column separation. In calculating the column separation we should also consider the distance to the complement of a column as well since it gives the same split of the set of biometric hash bits.

$$H_c(\boldsymbol{t}) = \min_{m,n,m\neq n} \left\{ \sum_{i=1}^{K} |C(i,m) - C(i,n)|, \sum_{i=1}^{K} |1 - C(i,m) - C(i,n)| \right\}, \tag{5.7}$$

where $m, n \in \{1, \ldots, \ell\}$ and K denotes the number of users. Maximizing the column separation will increase the verification accuracy of the system by decreasing correlation between classification errors [101, 102] and makes the system more robust against attacks. We define $H(\boldsymbol{t}) = H_r(\boldsymbol{t}) + H_c(\boldsymbol{t})$ as the optimization criterion to maximize. Hence, we have to solve $\hat{\boldsymbol{t}} = \arg\max_{\boldsymbol{t}} H(\boldsymbol{t})$.

Since the user cannot give exactly the same biometric template for each attempt to enter the system due to sensor imperfections and/or inherent user dependent variability, errors occur in biometric hashes. To decrease such errors, Hamming distance between the biometric hash vectors belonging to different users should be maximized. Besides, Hamming distance between each bit position of the biometric hash vectors should be maximized to reduce redundancy and to increase security against attacks.

We proceed as follows to optimize this complex objective. Initially, we find an optimum system level quantization threshold $\hat{t_s} \in [0, 1]$ that maximizes $H(t_s) = H_r(t_s) + H_c(t_s)$ by using the Golden section search (GSS) algorithm [103] in the range $[0, 1]$. The optimum system level threshold, $\hat{t_s} = \arg\max_{t_s} (H(t_s))$, is a quantization threshold which can be used for all bit positions of the biometric hash vectors.

Next, using the optimal system level threshold as an initial value, we find an optimum threshold for each bit position (m) of the biometric hash vectors that maximizes $H(\boldsymbol{t}) = H_r(\boldsymbol{t}) + H_c(\boldsymbol{t})$ by using the Golden section search algorithm [103] within the range $[0, 1]$. We perform this by using the coordinate descent method where we update one coordinate at a time while keeping the rest of the threshold vector constant. So, at each iteration, we solve optimization problem $\hat{t}(m) = \arg\max_{t(m)} (H(\tilde{\boldsymbol{t}}))$ for $m = 1, \ldots, \ell$ where $\tilde{\boldsymbol{t}} \in \mathbb{R}^{1\times\ell}$ is the latest threshold vector which contains the latest values of the thresholds for all bit positions and $\hat{t}(m) \in [0, 1]$ is the optimum threshold value for m^{th} bit position of the biometric hash vector. We go through all the coordinates multiple times until the iterations stop changing the objective value $H(\boldsymbol{t})$. The vector obtained in the end is the optimal bit-adaptive threshold vector $\hat{\boldsymbol{t}}$.

The pseudo-code of the optimization algorithm performed in the enrollment phase is given as Algorithm 1.

5.2.1.4 Relation with ECOC classification

In our scheme, we employ the column and row separation criteria used in ECOC matrix design to optimize the codeword matrix obtained from the biometric hash vectors. So, we do not pre-specify the codeword matrix and design classifiers afterward as regularly done in ECOC classification. Random projection followed by binary quantization that is used in biometric hashing can be seen as using a set of random linear classifiers $\boldsymbol{w}^T\boldsymbol{x} - b \gtrless 0$ where $\boldsymbol{w}$ corresponds to a single row of the random projection matrix and the bias term b corresponds to the bit-specific threshold t. Our method can be seen as using a number of random linear classifiers and modifying their bias value (the threshold) to optimize the row and column separation obtained by them. So, our method is not a direct application of ECOC multi-class classification, rather an innovative idea where random linear classifiers are optimized in their bias terms to obtain a better codeword matrix that will result in better verification performance.

5.2.2 Authentication Stage

At this stage, a claimer sends his face image $\tilde{\mathbf{I}} \in \mathbb{R}^{m\times n}$ and his secret key to the system. Then, the system computes the claimer's test biometric hash vector by using the same procedures in the enrollment phase with the optimum threshold for each bit position, $t(m)$. Finally, the system computes the Hamming distance [95] between the test biometric hash vector and the claimed

Algorithm 1 Pseudo Code of the Enrollment Phase

K : number of users and L : number of face images per user
ℓ : Length of biometric hash vector
Inputs: Training face images, $\mathbf{I}_{i,j}$, and secret keys of the users
Outputs: The binary codeword matrix, $\boldsymbol{C}$, and threshold vector $\boldsymbol{t}$
Compute PCA matrix $\mathbf{A}$ by using all the training images $\mathbf{I}_{i,j}$
for $i \leftarrow 1$ **to** K **do**
    Generate RP matrix $\mathbf{R}_{GS,i}$ by using the secret key of the i^{th} user
    for $j \leftarrow 1$ **to** L **do**
        Compute PCA coefficient vectors $\boldsymbol{x}_{i,j}$
        Compute $\boldsymbol{y}_{i,j} = \mathbf{A}(\boldsymbol{x}_{i,j} - \boldsymbol{\mu})$
        Compute $\boldsymbol{z}_{i,j} = \mathbf{R}_{GS,i}\boldsymbol{y}_{i,j}$
    end for
end for
for $i \leftarrow 1$ **to** K **do**
    for $m \leftarrow 1$ **to** ℓ **do**
        Compute $\boldsymbol{E}_i(m) = \frac{1}{L}\sum_{j=1}^{L} \boldsymbol{z}_{i,j}(m)$
    end for
end for
for $i \leftarrow 1$ **to** K **do**
    for $m \leftarrow 1$ **to** ℓ **do**
        Compute $\boldsymbol{V}_i(m) = \frac{E_i(m) - \min(\boldsymbol{E}_i)}{\max(\boldsymbol{E}_i) - \min(\boldsymbol{E}_i)}$
    end for
end for
$t_s^0 \leftarrow 0.5$ (set initial value of quantization threshold)
Solve $\hat{t_s} = \arg\max_{t_s} H_r(t_s) + H_c(t_s)$ using GSS algorithm
$t^0(m) = \hat{t_s}, m = 1, \ldots, \ell$ (set initial value of the threshold vector to the system level threshold)
Solve $\hat{\boldsymbol{t}} = \arg\max_{\boldsymbol{t}} H_r(\boldsymbol{t}) + H_c(\boldsymbol{t})$ using coordinate descent and GSS algorithm for each coordinate
Compute codeword matrix $\boldsymbol{C}$ by using the optimal threshold vector $\hat{\boldsymbol{t}}$
Store binary codeword matrix $\boldsymbol{C}$ and the threshold vector $\hat{\boldsymbol{t}}$

user's reference biometric hash vector stored in the database. If the Hamming distance is below the pre-determined distance threshold, the claimer is accepted; otherwise, the claimer is rejected.

Since the biometric hash vectors can only be computed by the system, a user typically does not know her biometric hash vector. Whenever a new user wants to enroll in the proposed system, the threshold vector, $\boldsymbol{t}$, and the codeword matrix, $\boldsymbol{C}$, which contains reference biometric hash vectors, stored in the database need to be updated. Initially, for new users, the system can use the existing threshold vector to determine their biometric hash vectors. When the number of new users reach a pre-defined specific number, the system will update itself (e.g. at night time when the system is idle) and generate a new threshold vector and a new codeword matrix which are optimal for the new population.

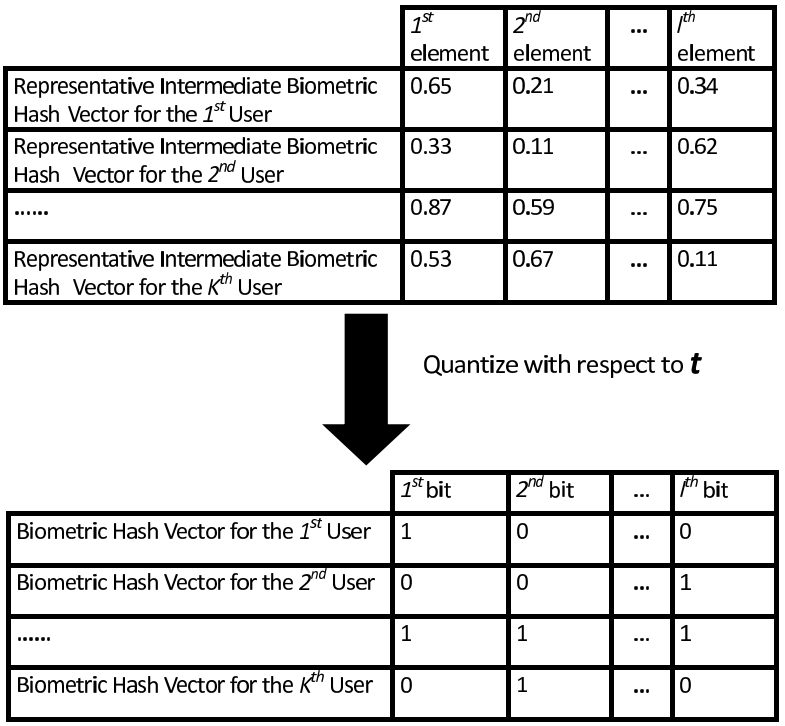

FIGURE 5.2: The illustration of the ECOC guided quantization step in the proposed biometric hashing scheme

In the proposed system, only the threshold vector, $\boldsymbol{t}$, is stored additionally in comparison with the Ngo *et. al.*'s system [2]. It is used for computing biometric hash vectors at the test stage. Even if an attacker obtains it, he cannot get any more information since the security of the system depends on the RP matrix and secret key of the users as in Ngo *et. al.* 's system [2]. The attacker can obtain neither the feature vector nor the biometric template of the user by using the threshold vector, $\boldsymbol{t}$, and the codeword matrix, $\boldsymbol{C}$, since there are infinitely many choices when getting back from binary biometric hash to the face image.

5.3 Simulation Results

In this section, we test and discuss the performance of the proposed scheme on Carnegie Mellon University (CMU) face database [98], Cambridge university AT&T face database [91], Multi Modal Verification for Teleservices and Security applications (M2VTS) face database [92, 93], and the Sheffield (previously UMIST) face databases [97]. Pre-processing methods such as eye marking, alignment and head region masking are not applied to the face images. Table 5.1 shows the number of face images used in the enrollment and test phases.

TABLE 5.1: Databases and experimental set-up

Database	Number of face images	Enrollment stage	Authentication stage
CMU	975 images from 13 people	The first 15 images of each user	The following 30 images of each user
AT&T	400 images from 40 people	The first 5 images of each user	The remaining 5 images of each user
M2VTS	1480 images from 37 people	The first 20 images of each user	The remaining 20 images of each user
Sheffield	564 images from 20 people	The first 8 images of each user	The following 8 images of each user

Table 5.2 shows the total number of genuine and imposter pairs. Note that all enrollment images for a person are used to generate a single reference biometric hash vector for a person. Each test image indicated in Table 5.1 is used once in a genuine test and as an impostor for all other users.

TABLE 5.2: Genuine and imposter pairs in each database

Database	Number of genuine pairs	Number of imposter pairs
CMU	$1 \times 30 \times 13 = 390$	$1 \times 30 \times ((12 \times 13) \div 2) = 2340$
AT&T	$1 \times 5 \times 40 = 200$	$1 \times 5 \times ((39 \times 40) \div 2) = 3900$
M2VTS	$1 \times 20 \times 37 = 740$	$1 \times 20 \times ((36 \times 37) \div 2) = 13320$
Sheffield	$1 \times 8 \times 20 = 160$	$1 \times 8 \times ((19 \times 20) \div 2) = 1520$

5.3.1 Equal Error Rate (EER) Performances

In this part, we test the performance of the proposed scheme. Kong *et al.* state that if unauthorized people steal the secret key and the RNG, the performances of biometric hashing schemes get worse [17]. Therefore, we simulate two scenarios, in our experiments as shown in Table 5.3.

1. Key-unknown Scenario: An imposter user wants to impersonate a genuine user. However, she has neither the biometric template nor the secret key of the user. She sends her own biometric template and a secret key to the system to be authenticated as the genuine user. In tests, we have used each impostor's own key for their impostor attempts as well.

2. Key-stolen Scenario: An imposter user obtains secret key of the genuine user. She sends her own biometric template and the secret key of the genuine user to the system to be authenticated as the genuine user.

As shown in Table 5.3, the proposed scheme has lower EER in comparison with [2]. We show the detection error trade-off (DET) curves [35] of the proposed method for key-stolen scenario

TABLE 5.3: EER performance comparison between the proposed biometric hashing scheme and Ngo *et al.*'s scheme [2]

Length	EER (%) of Ngo *et al.*'s scheme [2] (PCA+RP)	EER (%) of the proposed scheme	Scenario	Database
64 bit	% 2.05	**% 0.15**	Key-unknown	CMU
128 bit	% 0.98	**% 0.00**	Key-unknown	CMU
256 bit	% 0.60	**% 0.00**	Key-unknown	CMU
512 bit	% 0.22	**% 0.00**	Key-unknown	CMU
64 bit	% 4.10	**% 2.50**	Key-stolen	CMU
128 bit	% 2.46	**% 0.74**	Key-stolen	CMU
256 bit	% 1.72	**% 0.08**	Key-stolen	CMU
512 bit	% 1.18	**% 0.07**	Key-stolen	CMU
64 bit	% 12.19	**% 6.44**	Key-unknown	AT&T
128 bit	% 7.36	**% 4.25**	Key-unknown	AT&T
256 bit	% 5.81	**% 1.13**	Key-unknown	AT&T
512 bit	% 3.79	**% 0.04**	Key-unknown	AT&T
64 bit	% 16.93	**% 13.07**	Key-stolen	AT&T
128 bit	% 13.97	**% 9.71**	Key-stolen	AT&T
256 bit	% 12.76	**% 7.64**	Key-stolen	AT&T
512 bit	% 12.34	**% 8.01**	Key-stolen	AT&T
64 bit	% 18.10	**% 10.01**	Key-unknown	M2VTS
128 bit	% 14.56	**% 7.55**	Key-unknown	M2VTS
256 bit	% 11.15	**% 6.38**	Key-unknown	M2VTS
512 bit	% 9.23	**% 5.81**	Key-unknown	M2VTS
64 bit	% 21.36	**% 14.17**	Key-stolen	M2VTS
128 bit	% 18.08	**% 13.01**	Key-stolen	M2VTS
256 bit	% 16.72	**% 10.50**	Key-stolen	M2VTS
512 bit	% 16.09	**% 10.00**	Key-stolen	M2VTS
64 bit	% 17.09	**% 12.30**	Key-unknown	Sheffield
128 bit	% 16.38	**% 7.87**	Key-unknown	Sheffield
256 bit	% 15.05	**% 6.25**	Key-unknown	Sheffield
512 bit	% 14.97	**% 2.93**	Key-unknown	Sheffield
64 bit	% 21.40	**% 17.74**	Key-stolen	Sheffield
128 bit	% 21.92	**% 16.91**	Key-stolen	Sheffield
256 bit	% 22.53	**% 15.25**	Key-stolen	Sheffield
512 bit	% 23.47	**% 14.21**	Key-stolen	Sheffield

in Figures 5.3-5.6. In addition, we show genuine-imposter distance histograms and false accept rate (FAR) - false reject rate (FRR) plots in key-stolen scenario for the proposed method for the AT&T database for 64 and 128 bits in Figures 5.7-5.10. We attribute the performance improvements to the better scattering of biometric hash vectors due to the maximization of row and column separation in the codeword matrix in our method.

The proposed method more dramatically reduces the errors as the length of the biometric hash vector increases as shown in Table 5.3. The proposed scheme approximately reduces the EER by half in most of the cases. Furthermore, even in some cases, the proposed scheme perfectly separates the genuine and imposter users with no errors.

The proposed method maximizes the Hamming distance between the biometric hashes belonging to the different users at the enrollment stage. Thus, we achieve lower EERs in comparison with [2]. Even in the key-stolen scenario, we see improvements in performance which is possibly due to better placement of reference biometric hash vectors in the space of all possible hashes.

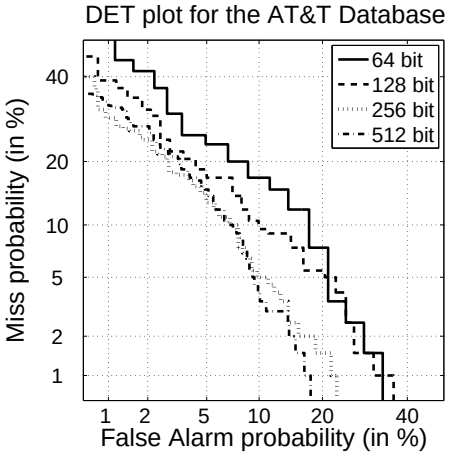

FIGURE 5.3: DET plots of the proposed method for key-stolen scenario - AT&T database

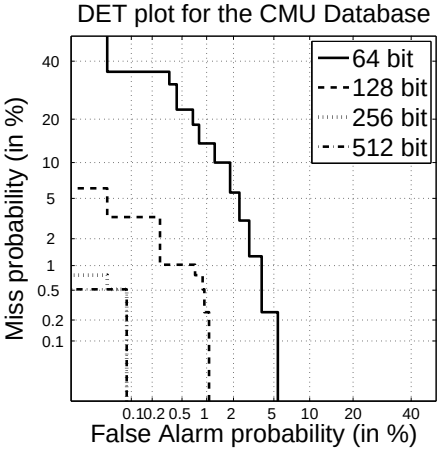

FIGURE 5.4: DET plots of the proposed method for key-stolen scenario - CMU database

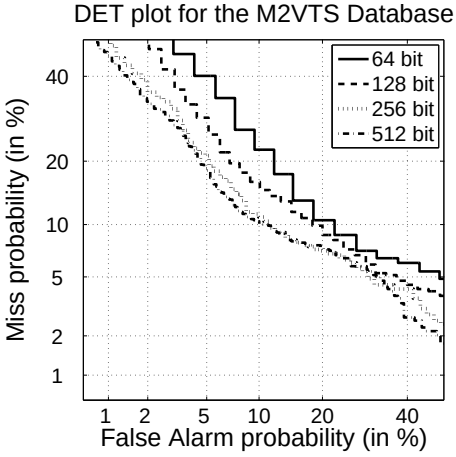

FIGURE 5.5: DET plots of the proposed method for key-stolen scenario - M2VTS database

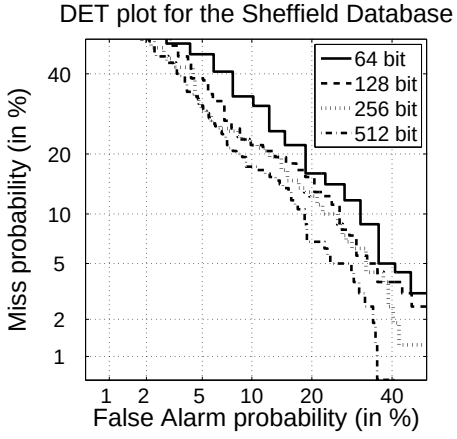

FIGURE 5.6: DET plots of the proposed method for key-stolen scenario - Sheffield database

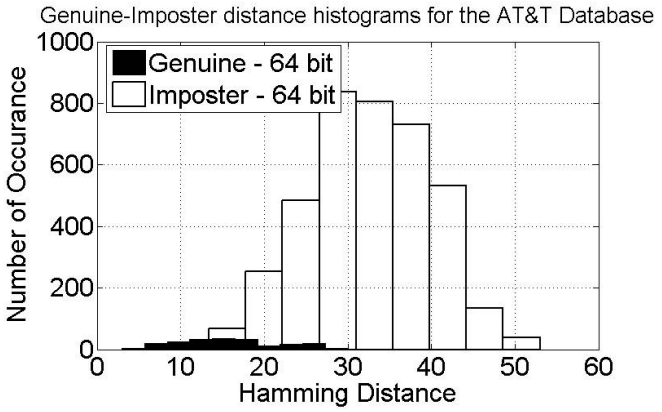

FIGURE 5.7: Genuine-Imposter distance histograms of the proposed method for key-stolen scenario in the AT&T database - 64 bit

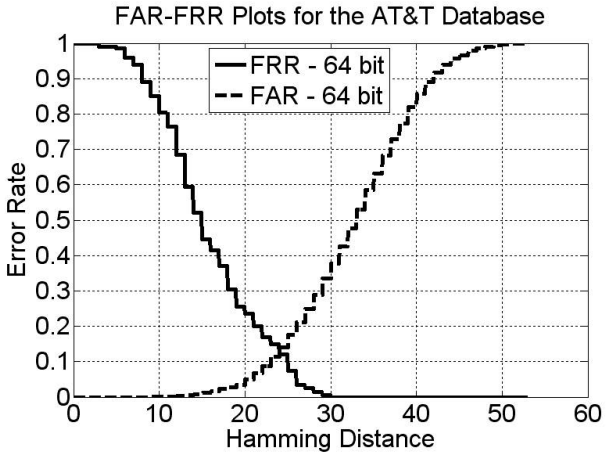

FIGURE 5.8: FAR-FRR plots of the proposed method for key-stolen scenario in the AT&T database - 64 bit

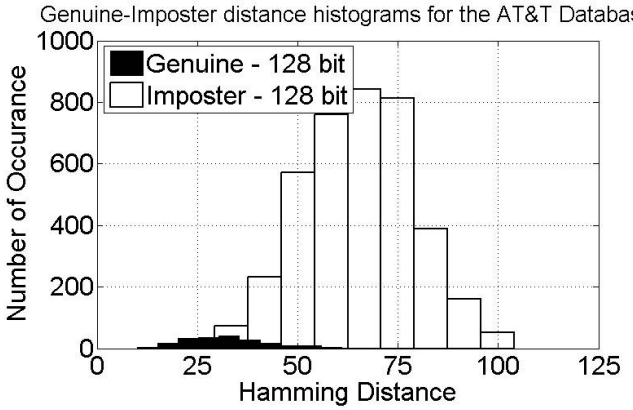

FIGURE 5.9: Genuine-Imposter distance histograms of the proposed method for key-stolen scenario in the AT&T database - 128 bit

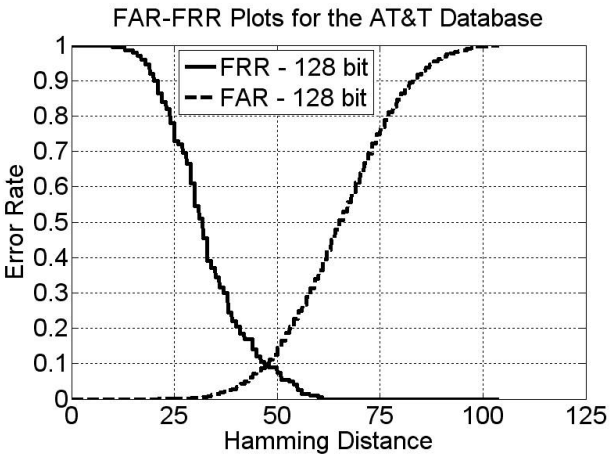

FIGURE 5.10: FAR-FRR plots of the proposed method for key-stolen scenario in the AT&T database - 128 bit

5.4 Comparison of the ECOC Guided Quantization For Biohashing and the Discriminative Biohashing Methods

In this thesis, we propose new biohashing methods by improving some phases of the random projection based methods. We develop new dimension reduction and quantization methods and propose the discriminative biohashing method in Chapter 4. Next, we design a new quantization method and propose the ECOC guided quantization for biohashing method in Chapter 5. Our aim is to improve the verification performance of the random projection based methods. In this section, we compare the performance of the ECOC guided quantization for biohashing and the discriminative biohashing methods. The comparison of the simulation results can be found in Table 5.4. The simulation results show that the ECOC guided quantization for biohashing method performs better than the discriminative biohashing method in most of the cases, especially when the bit-length of the hash vector is higher than 128. We conclude that the ECOC guided quantization for biohashing finds better quantization threshold values and thus improves the performance of the system. It appears that having better quantization thresholds is more preferable to using more discriminative projections.

TABLE 5.4: Comparison of the EER performances of the proposed biohashing methods in chapter 4 and chapter 5

Length	EER (%) of the ECOC guided quantization for biohashing [28]	EER (%) of the discriminative projection selection based biohashing [26] with binary quantization	EER (%) of the discriminative projection selection based biohashing [26] with GMM quantization	Scenario	Database
64 bit	% 6.44	% 3.83	% 2.73	Key-unknown	AT&T
128 bit	% 4.25	% 2.23	% 1.57	Key-unknown	AT&T
256 bit	% 1.13	% 1.80	% 1.15	Key-unknown	AT&T
512 bit	% 0.04	% 2.48	% 2.10	Key-unknown	AT&T
64 bit	% 13.07	% 14.40	% 13.58	Key-stolen	AT&T
128 bit	% 9.71	% 12.01	% 11.14	Key-stolen	AT&T
256 bit	% 7.64	% 10.80	% 10.23	Key-stolen	AT&T
512 bit	% 8.01	% 10.15	% 9.73	Key-stolen	AT&T
64 bit	% 12.30	% 15.75	% 16.23	Key-unknown	Sheffield
128 bit	% 7.87	% 13.33	% 14.03	Key-unknown	Sheffield
256 bit	% 6.25	% 11.45	% 11.05	Key-unknown	Sheffield
512 bit	% 2.93	% 10.44	% 12.20	Key-unknown	Sheffield
64 bit	% 17.74	% 19.38	% 20.68	Key-stolen	Sheffield
128 bit	% 16.91	% 17.51	% 19.71	Key-stolen	Sheffield
256 bit	% 15.25	% 16.96	% 17.80	Key-stolen	Sheffield
512 bit	% 14.21	% 19.27	% 18.22	Key-stolen	Sheffield

5.5 Chapter Summary

In this chapter, we propose a novel biometric hashing scheme based on the proposed quantization method that maximizes the row and the column separation of the code matrix as in ECOC classifiers. We maximize the distance between the genuine-impostor pairs as well as decrease the correlation between the bit positions in biometric hash vectors belonging to different users. The proposed method has superior performance in comparison with [2]. The proposed quantization method can be applied to other biometric hashing schemes that employ various feature extraction techniques.

Chapter 6

Security and Privacy Attacks Against Biohashing Schemes

6.1 Introduction

Biohashing scheme is recently proposed as a promising template protection method in the literature [2, 3, 16, 19–21]. Biohashing schemes offer to preserve privacy via randomization and enhance security via its cancelability property. Although these schemes achieve high performance in terms of authentication error rates and offer lower authentication times due to their binary nature, they are vulnerable to various attacks reported in the literature [17, 22–24].

Kong *et al.* [17] state that the performance of the biohashing schemes depend on the secrecy of the user's secret key and the actual performance of these methods should be evaluated under key-stolen attack scenario where an attacker gets the secret key of the legitimate user but does not have her face image. In this scenario, an attacker sends a face image and the secret key of the user to the system. Then, the system computes a biohash vector by using the data sent by the impostor. Finally, the system determines whether the claimer is authentic or not. Key-stolen scenario and its related attacks are addressed in previous chapters of this thesis.

In the literature, Cheung *et al.* [23] consider a security attack against the random projection based biohashing method [2]. The considered security attack can be seen as a simplified version of one of the security attacks considered in this chapter. They use the pseudo-inverse operator to approximately invert the random projection operation. However they assume that a

reconstructed biohash can be provided to the verification system directly for verification without forming a face image which may not be realistic in all cases. In addition, they do not consider privacy attacks in their paper. In another study, Lee *et al.* [104] point out that, the attacker does not need to know the user's secret key, just knowing the biohash is enough to come up with a pre-image attack where one can use any random projection matrix to obtain a feature vector which would produce the exact same biohash as the one that is discovered. However, such security attacks can be easily prevented by checking whether the user's secret key is entered correctly before comparing the biohashes. Therefore we do not consider such attacks in this chapter. Similar to Cheung *et al.* [23], Lee *et al.* [104] do not consider privacy attacks as well.

Other attacks against biohashing methods are performed by Kümmel *et al.* [22, 24] in the literature. They assume that an attacker steals some biohash vectors (in this case handwriting data) from the user and he tries to reconstruct the raw biometric data by using a genetic algorithm or a spline interpolation function. In these attacks, they need more than one biohash vector generated from the same biometric data as well as the secret key of the user. They try to perform a pre-image like attack. Their main aim is to threaten the security not the privacy since they do not want to reconstruct visually identical or similar handwriting instead they want to reconstruct a biometric data which can create a high matching score. In fact, it is almost impossible to reconstruct visually identical or similar biometric data due to huge information loss at the quantization phase of the biohashing methods.

In this chapter, we consider various attack scenarios that can threaten the privacy of the users as well as the security of the system. We perform attacks against random projection based biohashing methods [2, 3]. Since the feature extraction and random projection steps are dimension reducing linear transformations, there exists infinitely many elements which give the same result. That is, the inverse image of a single point under a linear dimension reduction operator is of infinite size. However it is easy to obtain a single element from this inverse image set using a minimum norm solution which is easy to obtain. The proposed attacks can also be performed for the biohashing systems that we propose in Chapter 3, Chapter 4, and Chapter 5. In Chapter 3, we propose an optimal linear transform based biohashing method. We develop a new dimension reduction method which results in a better dimension reduction matrix but we again use a matrix in the dimension reduction phase. In this method, we use exactly the same building blocks of a biohashing method: 1) Feature extraction, 2)Dimension reduction, and 3)Quantization. Eventually, the proposed attack method can be successfully performed for this method. In Chapter 4, we propose a discriminative biohashing method. We develop a new dimension

reduction method which selects the rows of the random projection matrix by applying the Fisher criterion and the indices of the selected rows are stored in the database. When a claimer sends her face image and the secret key to the system, the system computes the biohash vector by using the index vector $\mathbf{c}_i$. In other words, the proposed method uses a dimension reduction matrix defined by the user's secret key as in the random projection based biohashing method although our method offer a better dimension reduction matrix in order to obtain better error rates. We also proposed GMM based quantization in this method but it will not be a problem for applying our proposed attack method since we use another person's face image in order to invert the quantization step and the thresholds will be accessible to the attacker. As a result, the proposed attack method can successfully be performed for this method as well. In Chapter 5, we propose ECOC guided quantization for biohashing. Thus, we improve the performance of the system by developing a new quantization method. The feature extraction and dimension reductions phases are exactly the same as in the random projection based biohashing method. Since we use another person's face image in order to invert the quantization phase and the thresholds will be available to the attacker, this allows us to use the proposed attack method also for this biohashing method. Consequently, the proposed attack method can also be applied to the variants of the random projection based biohashing methods even if they use any other quantization method.

6.2 Desired Properties of Biohashes

In this section, we address the desired properties of the biohashes. They are summarized as follows:

1. The biohash should be irreversible.

   When an attacker gets a biohash of a legitimate user, she should not compute the biometric data of this user. In other words, even if the attacker get biohash h, no information about the biometric data itself should be obtained due to the desirable property that biohashes are irreversible.

2. The biohash should be cancelable.

   A biohashing scheme needs two inputs;

   (a) Biometric image, $Im \in \mathbf{I}$ where $\mathbf{I}$ is a biometric image space,

   (b) Secret key, $Key \in \mathbf{K}$ where $\mathbf{K}$ is a key space.

By using these two inputs, a biohash can be computed as follows:

$$\boldsymbol{h} = b\left(Im, Key\right), \tag{6.1}$$

where $\boldsymbol{h}$ is the biohash and $b\left(.\right)$ is a biohashing function. Therefore, a legitimate user can change/update her biohash by only changing her secret key when an attacker gets her biohash.

3. The biohash should be robust against different biometric images belonging to the same user.

   In real world applications, the users cannot give exactly the same biometric data in each enrollment session. Thus, the acquired biometric data of the user changes across different enrollment sessions. Even in such cases, it is expected that the computed biohashes (with different biometric data acquired in different sessions) of the same user should be similar. In other words, the biohashes, which are computed from the same user's biometric data and secret key, should be close in some distance metric with high probability.

$$d\left(b\left(Im_1, Key\right), b\left(Im_2, Key\right)\right) \leq t, \tag{6.2}$$

   where $d\left(.\right)$ is the function that computes Hamming distance between two inputs, $b\left(.\right)$ is a biohash function that computes biohash by using its inputs, Im_1 and Im_2 are two biometric images that belong to the same user and acquired at different sessions, Key is the user's secret key and t is the distance threshold for the decision.

4. Biohash should be fragile to the biometric images which do not belong to the legitimate user.

   The biohashes are not strict as the traditional cryptographic hashes. Thus, a single bit change in the input does not create a huge change at the output. Although, the biohashing schemes are robust against some changes in the input, they should be sensitive to the content changes. When the content of the biometric image is changed (different biometric image belonging to any other person), its corresponding biohash value should be distant from the biohash of the corresponding legitimate user.

$$d\left(b\left(Im_1, Key\right), b\left(Im_2, Key\right)\right) > t, \tag{6.3}$$

where $d(.)$ is the function that computes Hamming distance between two inputs, Im_1 and Im_2 are two biometric images that do not belong to the same user, and t is the distance threshold for the decision.

6.3 Privacy Threats

In this section, we perform attacks against Ngo *et al.*'s biohashing scheme [2, 3] in order to test its privacy preservation capability via its irreversibility property. In this attack scenario, we act as an attacker and try to threaten the privacy of the users who uses Ngo *et al.*'s biohashing scheme.

We address privacy flaws of random projection based biohashing schemes. As mentioned in the beginning of this chapter, our attacks will be applicable to the introduced methods in Chapters 3, 4, and 5 as well.

The main motivation behind this work is to find the minimum norm solution among the elements which map to the target value in the inverse image of the target value under the linear transformation. Although Ngo *et al.* claim that their random projection based biohashing scheme is irreversible [2, 3], we introduce a biohashing reconstruction method by using the weakness in the linear transformation steps of Ngo *et al.*'s biohashing scheme.

6.3.1 Attacks on the Irreversibility Property

In this part, we introduce our proposed attack method based on the minimum norm solution for random projection based biohashing schemes for face images. This is a new attack which is proposed by us in the literature. Although there are some works on pre-image attacks against biohashing schemes [24, 104] in order to threaten the security, there is no work on privacy threats

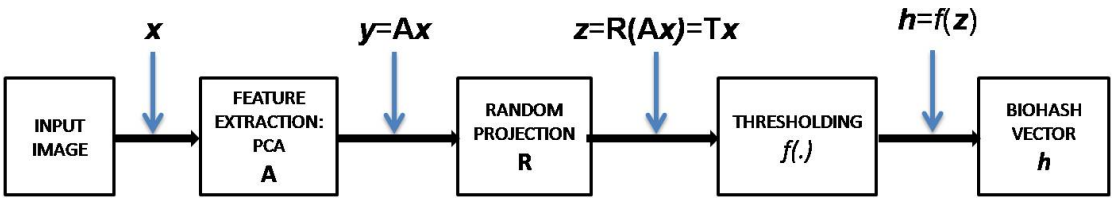

FIGURE 6.1: The basic steps of Ngo *et al.*'s biohashing scheme [2, 3]

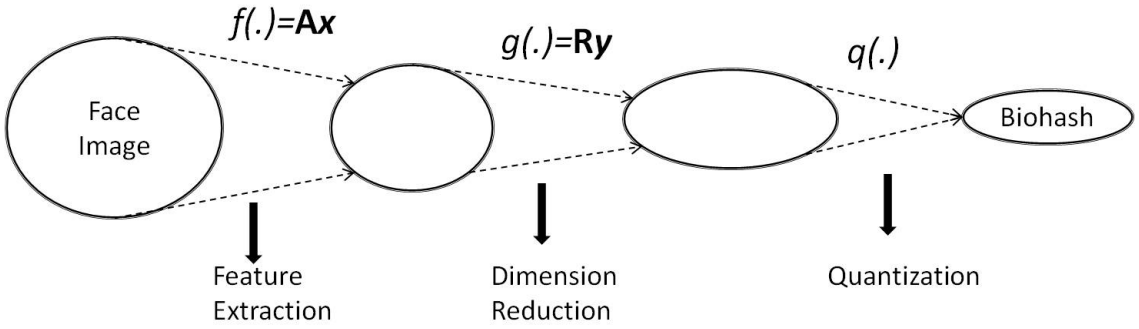

FIGURE 6.2: Illustration of Ngo *et al.*'s scheme's main phases in terms of functions

of biohashing schemes in the literature to the best of our knowledge. In this attack, we assume that the biohash vector, $\boldsymbol{h}$, of a legitimate user and her secret key are compromised. In this case, the attacker wants to recover the face image of the legitimate user by using the biohash vector and the secret key in order to threaten the privacy of the users. In other words, the attacker wants to create an artificial biometric data of the user by using her biohash. This artificially generated biometric data is desired to have the capability of generating false positive verifications although they are not necessary visually the same as the original biometric data. We can obtain a face image by inverting a biohash $\boldsymbol{h}$ as follows:

$$\hat{\mathbf{I}} = f^{\dagger}\left(g^{\dagger}\left(\tilde{q}\left(\boldsymbol{h}\right)\right)\right), \tag{6.4}$$

where $\hat{\mathbf{I}} \in \mathbb{R}^{m \times n}$ is the reconstructed face image, $\boldsymbol{h} \in \mathbb{R}^{\ell \times 1}$ is the biohash vector, $\tilde{q}\left(.\right)$, $g^{\dagger}\left(.\right)$, and $f^{\dagger}\left(.\right)$ denote a possible inverse function for the quantization, pseudo-inverse function for the random projection, and pseudo-inverse function for the feature extraction (i.e. Principle Component Analysis (PCA) [34]) phase in Ngo *et al.*'s biohashing method respectively.

Since the feature extraction and the random projection phases, which are illustrated in Figure 6.3, are linear transformations, we can find the elements which give the same results from their inverse image by using a minimum norm solution. Since Ngo *et al.*'s biohashing scheme reduces the dimensionality of the input face image in each step of their method, there exist infinitely many elements which maps to the same element in the inverse image.

6.3.1.1 The Proposed Attack Method Based on Minimum ℓ_2 Norm Solution

In this part, we introduce the proposed attack method based on the min-ℓ_2 norm solution. The details of the proposed attack method are given in the below part.

1. **Step 1:** The attacker steals a biohash vector, $\boldsymbol{h} \in \{0,1\}^{\ell}$, of a legitimate user either from a smart card or database etc.

2. **Step 2:** In Ngo *et al.*'s biohashing scheme, the elements of the intermediate biohash vector, z, are binarized with respect to a fixed threshold as follows:

$$\boldsymbol{h} = q(z) = \begin{cases} \boldsymbol{h}(i) = 1, & \text{if} \quad z(i) \geq t, \\ \boldsymbol{h}(i) = 0, & \text{Otherwise}, \end{cases} \tag{6.5}$$

   where $\boldsymbol{h} \in \{0,1\}^{\ell}$ is the biohash vector of the legitimate user and t is the mean value of the intermediate biohash vector $\boldsymbol{z}$.

   During this step, the attacker tries to simulate the inverse of the quantization phase in Ngo *et al.*'s biohashing scheme as follows:

$$\hat{\boldsymbol{z}} = \tilde{q}(\boldsymbol{h}) = \begin{cases} \hat{z}(i) = \alpha(i), & \text{if} \quad h(i) = 1, \\ \hat{z}(i) = \beta(i), & \text{if} \quad h(i) = 0, \end{cases} \tag{6.6}$$

   where $i = 1, \ldots, \ell$, $\tilde{q}(.)$ is the function for inverting the quantization phase of Ngo *et al.*'s biohashing scheme, and $\alpha(i)$ is a number larger than the threshold and $\beta(i)$ is a number less than the threshold. Since the threshold t would be unknown to the attacker, he needs to determine a threshold from another face image. Note that for methods which use a pre-determined threshold (such as the ones in Chapter 4 or 5), the attacker can obtain the threshold directly and act accordingly.

   The attacker uses another face image, which does not belong to the legitimate user, in order to perform this step of the proposed attack algorithm. He takes a face image, $\mathbf{I}_{fake} \in \mathbb{R}^{m \times n}$, which does not belong to any legitimate user. Next, he lexicographically re-orders it and obtains the face vector, $\boldsymbol{x}_{fake} \in \mathbb{R}^{(mn)\times 1}$. Then, he applies Principle Component Analysis (PCA) to the face vector as follows:

$$\boldsymbol{y}_{fake} = \mathbf{A}(\boldsymbol{x}_{fake} - \boldsymbol{w}), \tag{6.7}$$

where $\boldsymbol{y}_{fake} \in \mathbb{R}^{k\times 1}$ is the vector containing PCA coefficients of the face image which does not belong to the legitimate user.

Then, he projects these PCA coefficients onto lower dimensional space by using the secret key of the legitimate user.

$$\boldsymbol{z}_{fake} = \mathbf{R}_{GS}\boldsymbol{y}_{fake}. \tag{6.8}$$

Next, he maps the elements of the compromised biohash vector, $\boldsymbol{h}$, from $\{0, 1\}$ to $\{-1, 1\}$ space as follows:

$$\hat{\boldsymbol{h}} = \begin{cases} \hat{h}(i) = 1, & \text{if} \quad \boldsymbol{h}(i) = 1, \\ \hat{h}(i) = -1, & \text{if} \quad \boldsymbol{h}(i) = 0, \end{cases} \tag{6.9}$$

where $\hat{\boldsymbol{h}}$ is the mapped biohash vector.

The he computes the sample mean value of z_{fake}:

$$\mu = \frac{1}{\ell}\sum_{i=1}^{\ell} z_{fake}(i), \tag{6.10}$$

which would act as the threshold value t in the original system. Then, he computes the sample standard deviation of $\mathbf{z}_{fake}$:

$$\sigma = \left(\frac{1}{\ell}\sum_{i=1}^{\ell}\left(\boldsymbol{z}_{fake}(i) - \mu\right)^2\right)^{\frac{1}{2}}. \tag{6.11}$$

Finally, he computes $\hat{z}$ in the equation 6.6 as follows:

$$\hat{z}(i) = \mu + \sigma\left(\hat{h}(i)\right), \tag{6.12}$$

where $i = 1, \ldots, \ell$. That is, the attacker uses one standard deviation above or below the threshold value for inverting the quantization step.

3. **Step 3:** In Ngo *et al.*'s biohashing scheme, PCA feature vectors are projected by a random projection matrix which is generated with the user's secret key as follows:

$$\boldsymbol{z} = \mathbf{R}_{GS}\boldsymbol{y}, \tag{6.13}$$

where $\boldsymbol{z} \in \mathbb{R}^{\ell \times 1}$ is the randomly projected PCA feature vector (intermediate biohash vector), $\mathbf{R}_{GS} \in \mathbb{R}^{\ell \times k}$ is the orthonormal random projection matrix, and $\boldsymbol{y} \in \mathbb{R}^{k \times 1}$ is vector containing PCA coefficients of a legitimate user.

Since the attacker gets the secret key of the user, he can re-generate $\mathbf{R}_{GS}$ of the legitimate user. Then, he compute $\mathbf{R}_{GS}^{\dagger}$ by using the MoorePenrose pseudo-inverse method [1, 105, 106]. Therefore, the attacker performs inverse of the random projection phase with the help of compromised secret key and by taking into account Ngo *et al.*'s biohashing scheme as follows:

$$\hat{\boldsymbol{y}} = g^{\dagger}(\hat{\boldsymbol{z}}) = \left(\mathbf{R}_{GS}^{\dagger}\right)\hat{\boldsymbol{z}}, \tag{6.14}$$

where $g^{\dagger}(.)$ is the pseudo-inverse function for the random projection in Ngo *et al.*'s biohashing scheme, $\mathbf{R}_{GS} \in \mathbb{R}^{\ell \times k}$ is the orthonormal random projection matrix generated via the secret key compromised from the legitimate user, $\mathbf{R}_{GS}^{\dagger}$ is the pseudo-inverse matrix of $\mathbf{R}_{GS}$.

4. **Step 4:** In Ngo *et al.*'s biohashing scheme, first, a face image of the user, $\mathbf{I} \in \mathbb{R}^{m \times n}$, is lexicographically re-ordered and the face vector, $\boldsymbol{x} \in \mathbb{R}^{(mn) \times 1}$, is obtained. Then, PCA is applied to the face vector as follows:

$$\boldsymbol{y} = \mathbf{A}(\boldsymbol{x} - \boldsymbol{w}), \tag{6.15}$$

where $\mathbf{A} \in \mathbb{R}^{k \times (mn)}$ is the PCA matrix trained by the face images in the training set, $\boldsymbol{w}$ is the mean face vector, and $\boldsymbol{y} \in \mathbb{R}^{k \times 1}$ is vector containing PCA coefficients belonging to the face image of the user.

The attacker computes inverse of the PCA matrix and reconstruct the face image of the legitimate user.

$$\hat{\mathbf{I}} = f^{\dagger}(\hat{\boldsymbol{y}}) = \mathbf{A}^{\dagger}\hat{\boldsymbol{y}} + \boldsymbol{w}, \tag{6.16}$$

where $f^{\dagger}(.)$ is the pseudo-inverse function for the feature extraction (i.e. PCA) phase in Ngo *et al.*'s biohashing scheme, $\hat{\mathbf{I}}$ is the reconstructed face image, and $\mathbf{A}^{\dagger}$ is the pseudo-inverse of the PCA matrix.

6.3.1.2 The Proposed Attack Method Based on Minimum ℓ_1 Norm Solution

In this part, we introduce the proposed attack method based on the min-ℓ_1 norm solution. This attack is similar to the attack method based on minimum ℓ_2 norm solution introduced in Section 6.3.1.1. The only difference between them is in the third step. In other words, step 1,2 and 4 are the same in both attack methods. The step 3 for this attack method is explained in the following.

Step 3: In Ngo *et al.*'s biohashing scheme, PCA feature vectors are projected onto random projection matrix which is generated with the user's secret key as follows:

$$z = \mathbf{R}_{GS}\boldsymbol{y}, \tag{6.17}$$

where $\boldsymbol{z} \in \mathbb{R}^{\ell \times 1}$ is the randomly projected PCA feature vector (intermediate biohash vector), $\mathbf{R}_{GS} \in \mathbb{R}^{\ell \times k}$ is the orthonormal random projection matrix, and $\boldsymbol{y} \in \mathbb{R}^{k \times 1}$ is vector containing PCA coefficients of a legitimate user.

Since the attacker gets the secret key of the user, he can re-generate $\mathbf{R}_{GS}$ of the legitimate user. Min-ℓ_1 with equality constraints solution [107, 108], which is also known as basis pursuit, finds the vector with smallest ℓ_1 norm that explains the observations $\mathbf{z}$.

$$min \, \|\boldsymbol{y}\|_1 \quad \text{subject to} \quad z = \mathbf{R}_{GS}\mathbf{y}, \tag{6.18}$$

$$min \, \|\boldsymbol{y}\|_1 = \sum_i \left|\boldsymbol{y}_i\right|. \tag{6.19}$$

If there exists a sufficiently sparse $\boldsymbol{y}_0$ such that $\boldsymbol{z} = \mathbf{R}_{GS}\boldsymbol{y}_0$, then the equation 6.18 finds $\hat{\boldsymbol{y}}$ better at this step.

6.4 Security Threats

In cryptology, there are cryptographic hash functions which are the members of one-way functions. One-way function is a function that is easy to compute on every input; however, it is hard to invert given the image of a random input. Cryptographic hash functions take an arbitrary block of data and return a fixed size bit string. They are mostly used in content authentication,

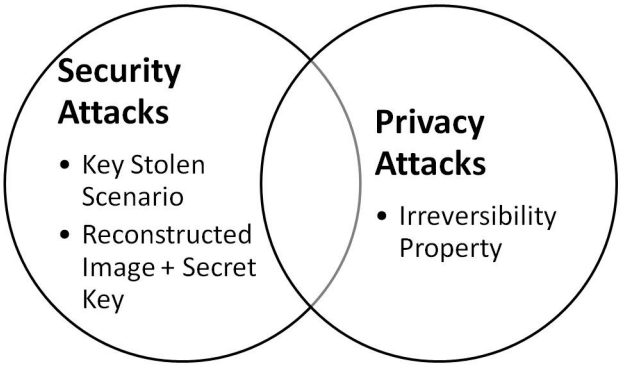

FIGURE 6.3: Security and privacy flaws of Ngo *et al.*'s scheme

search and indexing applications [109]. In the literature, there are several types of security constraints/properties on cryptographic hash functions. These functions are expected to be robust against the below type of attacks:

1. **Pre-Image Resistance**: Given a hash $\boldsymbol{h}$, it is hard to find any message $\boldsymbol{m}$ such that

$$\boldsymbol{h} = \text{hash}(\boldsymbol{m}), \tag{6.20}$$

   where *hash*(.) is the corresponding cryptographic hash function. This property comes from the one-wayness. In biohashing schemes, $\boldsymbol{m}$ is analogous to the biometric data of the user and $\boldsymbol{h}$ is analogous to the biohash vector.

2. **Second Pre-Image Resistance**: Given an input $\boldsymbol{m}_1$, it is hard to find another input $\boldsymbol{m}_2$, where $\boldsymbol{m}_1 \neq \boldsymbol{m}_2$ such that

$$\text{hash}\left(\boldsymbol{m}_1\right) = \text{hash}\left(\boldsymbol{m}_2\right). \tag{6.21}$$

   This property is sometimes referred as a weak collision resistance in the literature.

3. **Collision Resistance**: It is hard to find two different messages $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ such that

$$\text{hash}\left(\boldsymbol{m}_1\right) = \text{hash}\left(\boldsymbol{m}_2\right). \tag{6.22}$$

   This property is sometimes referred as a strong collision resistance in the literature.

Biohashing is totally different than cryptographic hashing. Cryptographic hash is a one-way function, when a single bit is changed the hash sum becomes completely different due to the avalanche effect [38]. On the other hand, for biohashing schemes several bits changes should not affect the performance of the system since they are designed by taking into account the noisy nature of the biometric templates. Therefore, we cannot expect from biohashing schemes to have exactly the same properties of cryptographic hashing. However, we use aforementioned properties in order to define attack models for biohashing schemes in Section 6.4.

6.4.1 Attacks on the Cancelability Property

In this part, we discuss possible attack scenarios on the cancelability property of the biohashing schemes. Let us assume that an attacker steals the biohash of a legitimate user. The legitimate user recognizes that her biohash has been stolen and she re-issues a new biohash vector by changing her secret key. In this case, the attacker wants to authenticate himself by using the old biohash of the legitimate user. In this case, we assume that providing a biohash directly to a verification system is possible.

6.4.2 Attack Scenarios for the Minimum ℓ_1 and Minimum ℓ_2 Norm Solution Based Attack Methods

In this part, we introduce other possible attack scenarios which uses the minimum ℓ_1 and minimum ℓ_2 norm solution based attack methods described in Section 6.3.1.2 and Section 6.3.1.1. These attacks can also threaten the security of the random projection based biohashing schemes and they are analogous to an attack against the pre-image resistance property of a cryptographic hash function described in Section 6.4 since they both try to reconstruct the face image of the legitimate user. Recall that pre-image resistance property states that given a hash $\boldsymbol{h}$, it is hard to find any message $\boldsymbol{m}$ such that $\boldsymbol{h} = \text{hash}(\boldsymbol{m})$. Consequently, we come up with three main attack scenarios which are described below.

1. **Scenario 0**: Consider a remote biometric verification system where a user and a verifier communicate via a communication channel and a biohashing scheme based verification is used. In this scenario, the verification is done via biohash values. From the security point of view, the attacker wants to cheat the system by getting a legitimate users biohash during the communication session or directly from the database. Thus, he gets both the

biohash and the secret key of a legitimate user. Then, he reconstructs the face image of the legitimate user by using the biohash and the secret key. Finally, he wants to enter the system by using the reconstructed face image and the secret key of the user.

2. **Scenario 1**: Consider a remote biometric verification system where a user and a verifier communicate via a communication channel and a biohashing scheme based verification is used. In this scenario, the verification is done via biohash values. From the security point of view, the attacker wants to cheat the system by getting a legitimate users biohash during the communication session or directly from the database. Thus, he gets the biohash of a legitimate user from this communication channel. He also gets the secret key of the user. Then, he reconstructs the face image of the legitimate user by using the biohash and the secret key. However, the user notices that her biohash was stolen and changes/updates her biohash by changing her secret key. Besides, the system administrator increase the security of the communication channel between the user and the verifier so that the attacker cannot enter this communication channel (i.e. use SSL (Secure Sockets Layer) connection). In this case, the attacker wants to authenticate himself to the system by using the reconstructed face image and the old secret key.

3. **Scenario 2**: In this scenario, an attacker again gets biohash of a legitimate user from the communication channel between the user and the verifier. He also acquires the secret key of the user. Then, he reconstructs the face image of the legitimate user by using the biohash and the secret key. However, the user notices that her biohash was stolen and changes/updates her biohash by changing her secret key. Besides, the system administrator increase the security of the communication channel between the user and the verifier so that the attacker cannot enter this communication channel (i.e. use SSL connection). On the other hand, the attacker again steals the secret key of the user. Therefore, the attacker wants to authenticate himself to the system by using the reconstructed face image and the new secret key. This attack scenario is very similar to the key-stolen attack scenario addressed in previous chapters. The main difference is that this attack scenario is performed by a skilled imposter since he uses the reconstructed face image of a legitimate user whereas the attacker uses another person's face image in the key-stolen attack scenario.

In addition to these attacks, it is obvious that biohashing systems inherently have verification errors. These verification errors may cause intrinsic failure attacks (i.e., zero-effort attacks, brute

force attacks) which are derived from the fact that there is always a non-zero probability that two biometric templates generated from two different individuals are sufficiently alike to produce a positive match [1]. Thus, an incorrect decision can be made by a biometric recognition system (e.g., a false accept). For instance, an attacker can try to authenticate herself to the system by using a biometric data and a secret key which do not belong to any legitimate user. It is analogous to an attack against the second pre-image resistance or collision resistance property of a cryptographic hash function described in Section 6.4. This attack scenario is covered as "key-stolen scenario" in other chapters of this thesis.

6.5 Simulation Settings and Results

In this section, we give the simulation settings and the results of the simulations for privacy and security attacks.

6.5.1 Simulations for the Privacy Threats: Attacks on the Irreversibility Property

In this part, we give the simulation settings and results of the attacks in order to test the irreversibility property of Ngo *et al.*'s random projection based biohashing scheme for face images [2]. In simulations, k is set to 1024 for $\boldsymbol{y} \in \mathbb{R}^{k\times 1}$ which is the vector containing PCA coefficients belonging to the face image of the user. Besides, ℓ, which is the length of a biohash vector, is set to $\{64, 128, 256, 512\}$.

The reconstructed, original and mean face images are shown in Figure 6.4 - Figure 6.7. We recontruct visually similar face images by using min ℓ_2 and min ℓ_1 norm solutions with various biohash vector lengths. It is obvious that if noise removal filters and some image processing techniques are applied to the reconstructed face image, it is possible to obtain better images.

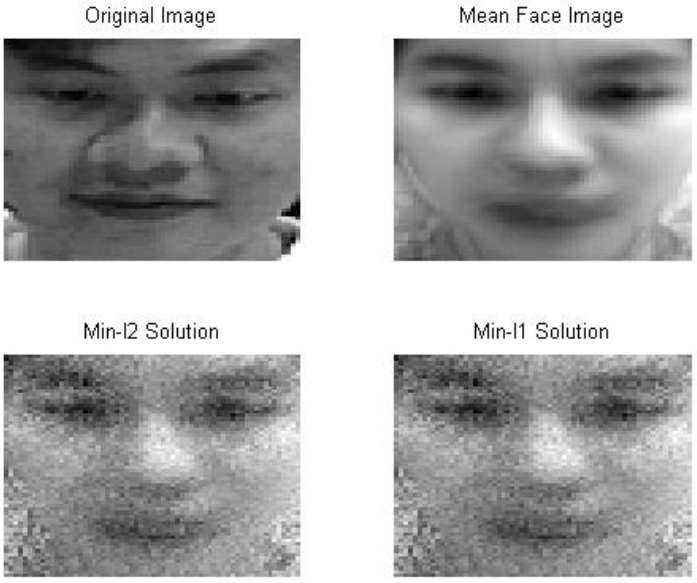

FIGURE 6.4: Illustration of the original image, mean face image and the reconstructed face images by using min ℓ_2 and min ℓ_1 norm solutions with 64 bit biohash vector.

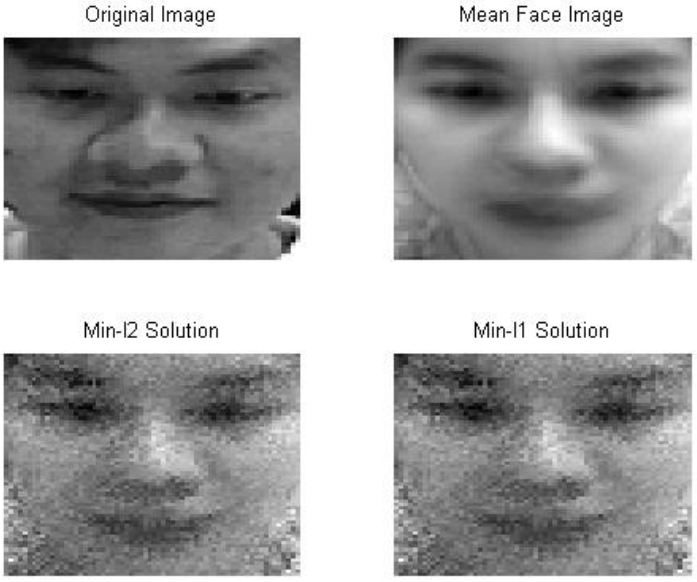

FIGURE 6.5: Illustration of the original image, mean face image and the reconstructed face images by using min ℓ_2 and min ℓ_1 norm solutions with 128 bit biohash vector.

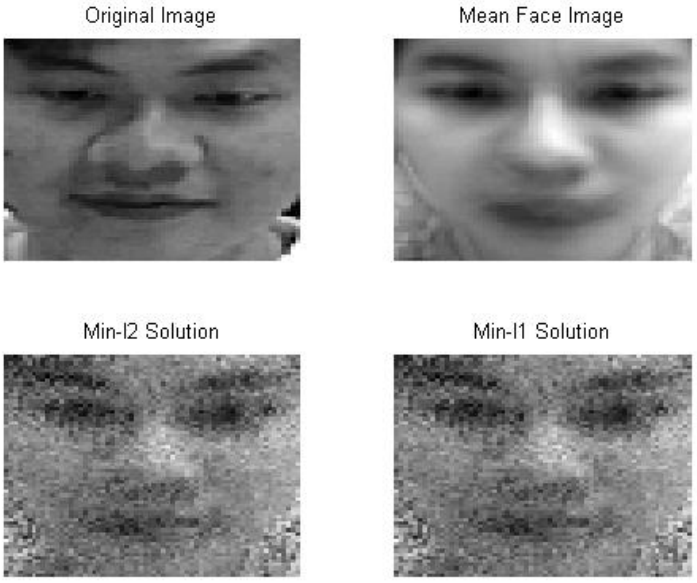

FIGURE 6.6: Illustration of the original image, mean face image and the reconstructed face images by using min ℓ_2 and min ℓ_1 norm solutions with 256 bit biohash vector.

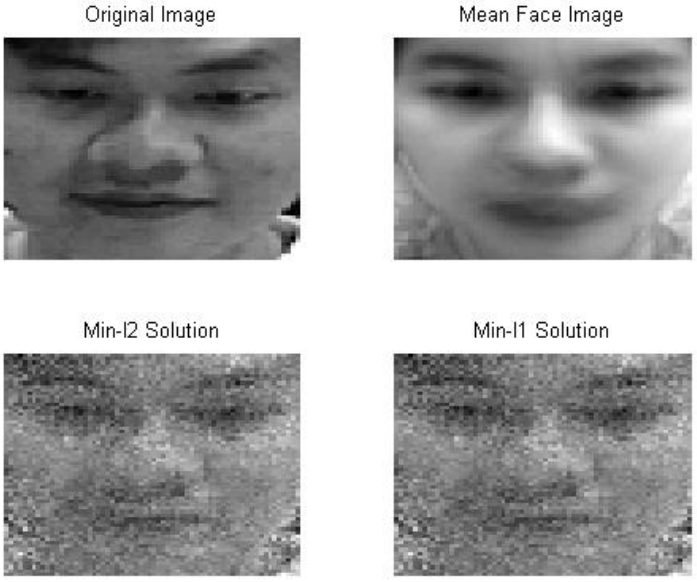

FIGURE 6.7: Illustration of the original image, mean face image and the reconstructed face images by using min ℓ_2 and min ℓ_1 norm solutions with 512 bit biohash vector.

6.5.2 Simulations for the Security Attacks

In this part, we give the simulation setting and results for the security attacks against Ngo *et al.*'s random projection based biohashing scheme for face images [2]. We perform experiments in order to demonstrate the general validity of attack models with various face databases. First, we demonstrate the performance of the proposed attack method on the Cambridge university AT&T face database [91]. In this database, there are 400 different face images from 10 different images of each of 40 distinct subjects. The images were taken at different times, varying the lighting, facial expressions and facial details. The size of the face images are $m = 112$ and $n = 92$. Next, we test the performance of the proposed attack method on the Multi Modal Verification for Teleservices and Security applications (M2VTS) face database [92]. The face images in this database are taken from the videos as explained in [93]. That database consists of face images with various expressions, illumination conditions, angles, age, sex, and glasses. The size of the face images are $m = 48$ and $n = 64$. There are 1480 face images, which consist of 40 different face poses for 37 different people in the database. Next, we test the performance of the proposed attack method on the Sheffield (previously UMIST) face databases [97]. The Sheffield database has 564 different face images belonging to 20 different people. Each people have various face images with wide range of poses from profile to frontal views. Finally, we test the performance of the proposed attack method on the AR database which has 3120 face images belonging to 120 different people's faces with different facial expressions, illumination conditions, and occlusions.

6.5.2.1 Simulations for the attacks against cancelability property

In this part, we perform simulations in order to test whether Ngo *et al.*'s random projection based biohashing scheme satisfies the cancelability property or not as defined in Section 6.4.1. We also perform random guess attacks where an attacker tries to guess the biohash of a legitimate user without knowing her new or any previous biohash vector. In this case, the attacker neither knows the biometric data nor the secret key of a legitimate user as well. We compare this two attack scenarios. The simulation results, which are shown in Figure 6.8-Figure 6.11, demonstrate that random projection based biohashing methods satisfy the cancelability property. In other words, the attacks against cancelability property are not better than the random guess attacks.

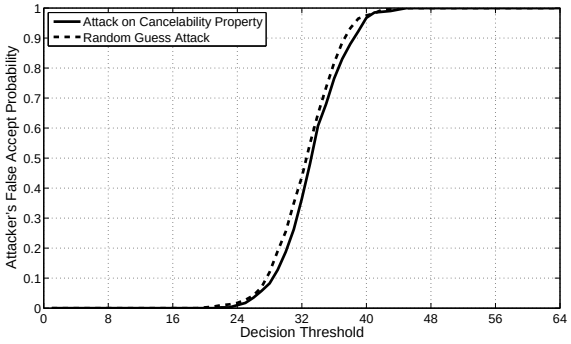

FIGURE 6.8: The change of false accept probability with respect to various decision thresholds for stolen biohash of 64-bit length on AT&T database.

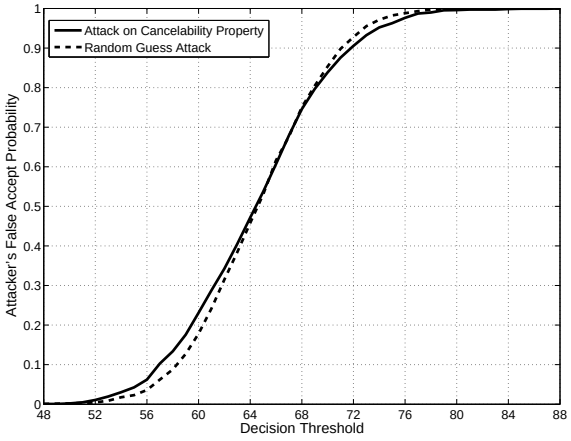

FIGURE 6.9: The change of false accept probability with respect to various decision thresholds for stolen biohash of 128-bit length on M2VTS database.

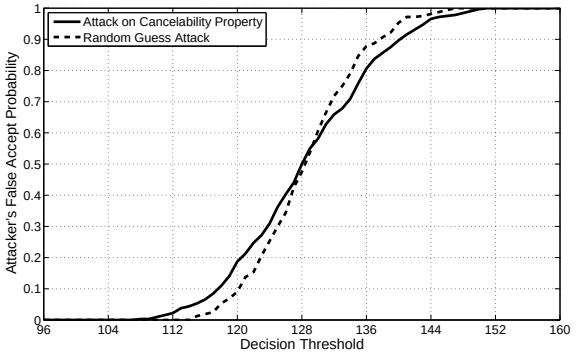

FIGURE 6.10: The change of false accept probability with respect to various decision thresholds for stolen biohash of 256-bit length on Sheffield database.

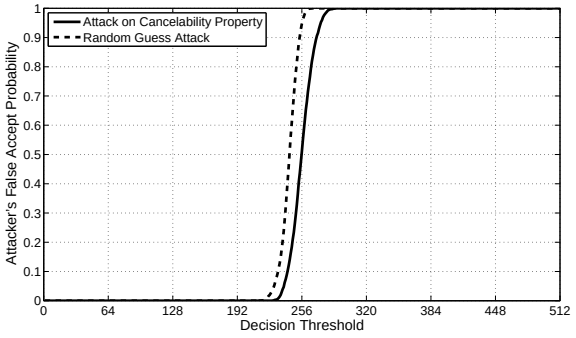

FIGURE 6.11: The change of false accept probability with respect to various decision thresholds for stolen biohash of 512-bit length on AR database.

6.5.2.2 Simulations for the attack scenarios using the minimum ℓ_1 and minimum ℓ_2 norm solution based attack methods

In this part, we provide the simulation setting and results for the attack scenarios, which are defined in Section 6.4.2, in Figure 6.16-Figure 6.15 and in Table 6.2-Table 6.1. In particular, the simulation results for the proposed attack based on min-ℓ_1 norm solution can be seen in Figure 6.16, Figure 6.17 and Table 6.2. Furthermore, the simulation results for the proposed attack based on min-ℓ_2 norm solution can be seen in Figure 6.14, Figure 6.15 and Table 6.1. The simulation results show the attack based on the min-ℓ_2 norm solution performs better than the attack based on the min-ℓ_1 norm solution. It is expected since the PCA feature vectors are not sparse enough.

It is obvious that when an attacker gets the biohash and the secret key of a legitimate user and reconstructs the face image of the user, she can enter the system successfully as long as the user is not aware that her biohash is stolen. This corresponds to the attack scenario 0. The simulation result of the min-ℓ_1 norm solution for 64 bit biohash vectors can be seen in Figure 6.12. In addition, the simulation result of the min-ℓ_2 norm solution for 512 bit biohash vectors can be seen in Figure 6.13. Note that, this simulation result can also be obtained for all lengths of biohash vector (64, 128, 256 and 512 bit) on all face databases (AT&T, Sheffield, M2VTS, AR) by using either the min-ℓ_1 norm solution or the min-ℓ_2 norm solution.

The proposed attack methods, especially, perform better (i.e. they yield higher EERs) with 256 and 512 bit biohash vectors. From the simulation results, we deduce that as the attacker gets more information either on biometric data or secret key, he performs more successful attacks. It

TABLE 6.1: EER performance of the proposed attack methods based on min-ℓ_2 norm solution against Ngo *et al.*'s method [2, 3]

Length	Key-unknown scenario	Key-stolen scenario	Scenario 0	Scenario 1	Scenario 2	Database
64 bit	% 12.19	% 16.93	% 100.00	% 11.97	% 22.78	AT&T
128 bit	% 7.36	% 13.97	% 100.00	% 7.12	% 21.93	AT&T
256 bit	% 5.81	% 12.76	% 100.00	% 5.30	% 26.58	AT&T
512 bit	% 3.79	% 12.34	% 100.00	% 3.61	% 45.16	AT&T
64 bit	% 17.09	% 21.40	% 100.00	% 21.38	% 30.69	Sheffield
128 bit	% 16.38	% 21.92	% 100.00	% 17.49	% 35.12	Sheffield
256 bit	% 15.05	% 22.53	% 100.00	% 17.97	% 43.36	Sheffield
512 bit	% 14.97	% 23.47	% 100.00	% 14.43	% 58.57	Sheffield
64 bit	% 18.10	% 21.36	% 100.00	% 12.20	% 21.30	M2VTS
128 bit	% 14.56	% 18.08	% 100.00	% 7.21	% 21.64	M2VTS
256 bit	% 11.15	% 16.72	% 100.00	% 5.69	% 27.48	M2VTS
512 bit	% 9.23	% 16.09	% 100.00	% 3.87	% 44.58	M2VTS
64 bit	% 23.63	% 28.27	% 100.00	% 24.95	% 32.50	AR
128 bit	% 18.24	% 27.17	% 100.00	% 21.95	% 32.20	AR
256 bit	% 15.82	% 25.50	% 100.00	% 12.45	% 45.86	AR
512 bit	% 11.38	% 24.89	% 100.00	% 15.30	% 61.03	AR

is concluded that the attacker is more successful in the second attack scenario than in the first attack scenario since he also knows the new secret key in this case.

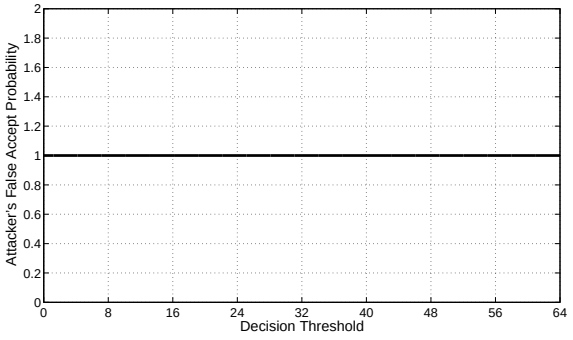

FIGURE 6.12: The change of false accept probability with respect to "attack scenario-0" defined in Section 6.4.2 for 64 bit biohash vector on all face databases (AT&T, Sheffield, M2VTS, AR) by using the min-ℓ_2 norm solution.

TABLE 6.2: EER performance of the proposed attack methods based on min-ℓ_1 norm solution against Ngo *et al.*'s method [2, 3]

Length	Key-unknown Scenario	Key-stolen Scenario	Scenario 0	Scenario 1	Scenario 2	Database
64 bit	% 12.19	% 16.93	% 100.00	% 12.60	% 15.91	AT&T
128 bit	% 7.36	% 13.97	% 100.00	% 8.58	% 13.96	AT&T
256 bit	% 5.81	% 12.76	% 100.00	% 5.66	% 16.50	AT&T
512 bit	% 3.79	% 12.34	% 100.00	% 4.00	% 32.37	AT&T
64 bit	% 17.09	% 21.40	% 100.00	% 19.42	% 25.54	Sheffield
128 bit	% 16.38	% 21.92	% 100.00	% 15.94	% 26.22	Sheffield
256 bit	% 15.05	% 22.53	% 100.00	% 15.28	% 34.82	Sheffield
512 bit	% 14.97	% 23.47	% 100.00	% 15.48	% 41.18	Sheffield
64 bit	% 18.10	% 21.36	% 100.00	% 11.87	% 16.00	M2VTS
128 bit	% 14.56	% 18.08	% 100.00	% 8.15	% 14.84	M2VTS
256 bit	% 11.15	% 16.72	% 100.00	% 5.32	% 17.37	M2VTS
512 bit	% 9.23	% 16.09	% 100.00	% 4.05	% 34.44	M2VTS
64 bit	% 23.63	% 28.27	% 100.00	% 26.55	% 20.33	AR
128 bit	% 18.24	% 27.17	% 100.00	% 15.51	% 22.40	AR
256 bit	% 15.82	% 25.50	% 100.00	% 13.75	% 23.23	AR
512 bit	% 11.38	% 24.89	% 100.00	% 12.23	% 37.44	AR

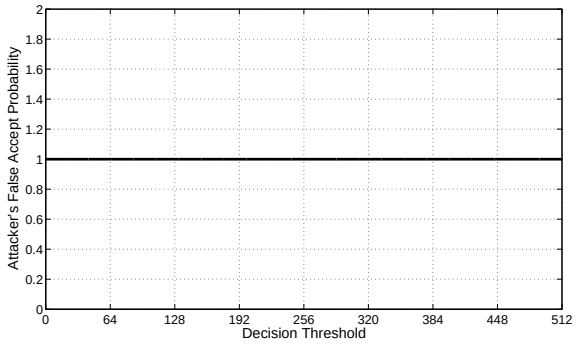

FIGURE 6.13: The change of false accept probability with respect to "attack scenario-0" defined in Section 6.4.2 for 512 bit biohash vector on all face databases (AT&T, Sheffield, M2VTS, AR) by using the min-ℓ_1 norm solution.

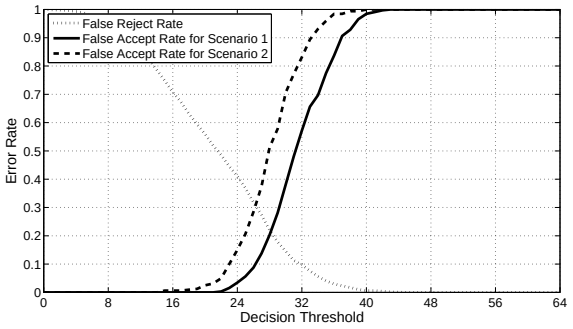

FIGURE 6.14: The change of FRR and FAR with respect to the decision threshold for "attack scenario-1" and "attack scenario-2" defined in Section 6.4.2 with 64 bit biohash vector on Sheffield database by using the min-ℓ_2 norm solution.

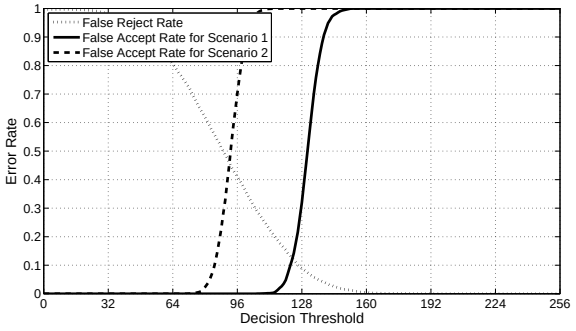

FIGURE 6.15: The change of FRR and FAR with respect to the decision threshold for "attack scenario-1" and "attack scenario-2" defined in Section 6.4.2 with 256 bit biohash vector on AR database by using the min-ℓ_2 norm solution.

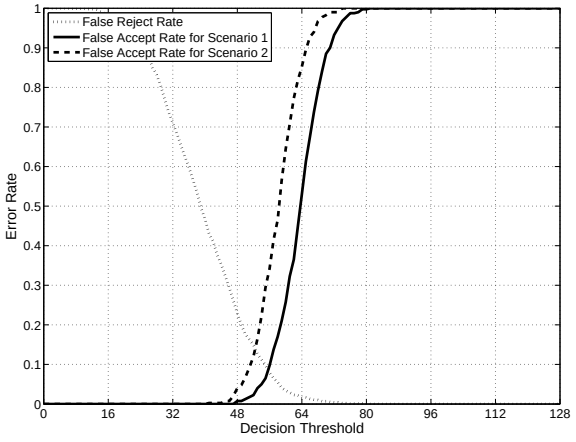

FIGURE 6.16: The change of FRR and FAR with respect to the decision threshold for "attack scenario-1" and "attack scenario-2" defined in Section 6.4.2 with 128 bit biohash vector on AT&T database by using the min-ℓ_1 norm solution.

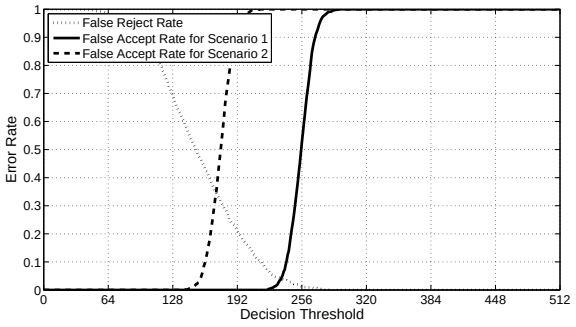

FIGURE 6.17: The change of FRR and FAR with respect to the decision threshold for "attack scenario-1" and "attack scenario-2" defined in Section 6.4.2 with 512 bit biohash vector on M2VTS database by using the min-ℓ_1 norm solution.

6.6 Chapter Summary

In this chapter, we categorize the threats into two groups: 1) Privacy threats, 2)Security threats. We introduce new privacy and security attacks against the random projection based biohashing schemes. We propose attacks based on minimum norm solutions of linear projections and a possible inversion of the quantization step. These attacks threaten the privacy of the users as well as the security of the system. The simulation results demonstrate that the random projection based biohashing methods do not satisfy the irreversibility property under some conditions and this makes them vulnerable against severe privacy and security attacks.

In addition to these attacks, there may be other attacks that can be performed against the random projection based biohashing methods. For instance, the attacker may use a face image database containing face images which do not belong to any legitimate user instead of reconstructing the face image by using the biohash and the secret key of the user. Thus, he can use the face image whose biohash has the minimum Hamming distance with the user's biohash.

Chapter 7

THRIVE: Threshold Homomorphic encRyption based secure and privacy preserving bIometric VErification system

7.1 Introduction

[1] Recently, homomorphic encryption methods are used with biometric feature extraction methods in order to perform verification via encrypted biometric templates [54, 84–86]. However, these methods offer solutions in the honest-but-curious model where each party is obliged to follow the protocol but can arbitrarily analyze the knowledge that it learns during the execution of the protocol in order to obtain some additional information. Their proposed system is not designed for the malicious model where each party can arbitrarily deviate from the protocol and may be corrupted. On the other hand, they do not take into account security and privacy issues of biometric templates stored in the database [54, 86]. The authors state that their security model will be improved in the future work by applying encryption methods also on the biometric templates stored in the database. Furthermore, some of these systems are just designed for a single biometric modality or a specific feature extraction method which limits their application areas

[1] This chapter is based on [29]. I would like to especially thank Dr. Mehmet Sabir Kiraz for his valuable contributions to this chapter especially on cryptography related parts.

[84, 85]. Apart from that, an adversary can enroll himself on behalf of any user to their systems since they do not offer any solutions for malicious enrollment. Finally, all these systems suffer from computational complexity.

Biohashing schemes are one of the emerging biometric template protection methods [16, 18–21]. These schemes offer low error rates and fast verification at the authentication stage. However, they suffer from several attacks reported attacks as shown in Chapter 6 and in the literature [17, 22–24]. These schemes should be improved in order to be safely used in a wide range of real life applications. In this chapter, we develop new enrollment and authentication protocols for biometric verification methods and we call it "THRIVE: Threshold Homomorphic encRyption based secure and privacy preserving bIometric VErification system". Our goal is to increase security and enhance privacy of the biometric schemes. The THRIVE system can work with any biometric feature extraction scheme whose outputs are binary or can be binarized. Since biohashing schemes can output binary templates called a biohash, they can be successfully used with the proposed system. We address adversary attacks (which are introduced in [4]) when a malicious attacker aims to gain access to the system. By taking into account these adversary attacks, we develop a new biometric authentication system based on threshold homomorphic encryption in the malicious model. Our general aim is to increase security of the system and enhance privacy of biometric templates belonging to the users of the system. The contributions of the THRIVE system introduced in this chapter can be summarized as follows:

1. A new biometric authentication system (which we call the THRIVE system) is proposed in the malicious model and the proposed system can be used with any existing biometric modality whose templates are fixed size vectors and they can be binarized.

2. Only a legitimate user can enroll in the proposed system since a signature scheme is used at the proposed enrollment stage.

3. Only encrypted versions of binary templates are stored in the database and biometric templates are never released even during authentication. Thus, the proposed system offers a new and advanced biometric template protection method without any helper data.

4. Even if an adversary gains an access to the database and steals encrypted biometric templates, neither he can authenticate himself by using these encrypted biometric templates due to the authentication protocol nor he can decrypt these encrypted biometric templates due to the $(2, 2)$-threshold homomorphic encryption scheme.

5. Neither the verifier nor the user can perform decryption by themselves on encrypted biometric templates since the $(2, 2)$-threshold homomorphic encryption scheme is used. Instead, the verifier and the user can perform decryption collaboratively using their own private key shares.

6. The verifier does not need to know the user's biometric template or private key in order to authenticate the user.

7. The THRIVE system can be used in the applications where the user does not trust the verifier since the user does not need to reveal her biometric template and/or private key in order to authenticate herself and the verifier does not need to reveal any data to the user at the proposed authentication protocol.

8. Authentication is performed via randomized templates which ensures privacy.

9. Even if an adversary intercepts the communication channel between the user and the verifier, he cannot obtain any useful information on the biometric template since all exchanged messages are randomized and/or encrypted and he cannot perform decryption due to the $(2, 2)$-threshold homomorphic encryption scheme. Furthermore, he cannot use the obtained data from message exchanges in this communication channel since nonce and signature schemes are used together in the authentication.

10. The THRIVE system is a two-factor authentication system (biometric and secret key) and is secure against illegal authentication attempts. In other words, a malicious adversary cannot gain access to the proposed system without having the biometric data and the private key of a legitimate user by performing adversary attacks described in [4] as well as hill-climbing attacks [30–33].

11. In the THRIVE system, the generated protected biometric templates are irreversible since they are encrypted.

12. In the THRIVE system, the generated protected biometric templates are cancelable. Even if they are stolen, they can be re-generated.

13. The THRIVE system can generate a number of protected templates from the same biometric data of a user due to the randomized encryption and biohashing. Thus, it ensures diversity.

14. The THRIVE system is implemented and a successful authentication protocol run requires 0.218 seconds on average. Consequently, the proposed system is sufficiently efficient to be used in real world applications.

7.2 Attacks on Biometric Systems

In the literature, attacks to a biometric system are classified into two categories [4]: 1) Intrinsic failure attacks, 2) Adversary attacks. These attacks are introduced in the below part. In these attacks, an adversary may have two main aims:

1. Threaten security by gaining access to the system,
2. Threaten privacy by obtaining the biometric data of a user.

7.2.1 Intrinsic failure attacks

Intrinsic failure attacks (i.e., zero-effort attacks, brute force attacks) [1] are derived from the fact that there is always a non-zero probability that two biometric templates generated from two different individuals are sufficiently alike to produce a positive match. Thus, an incorrect decision can be made by a biometric recognition system (e.g., a false accept).

7.2.2 Adversary attacks

Adversary attacks are caused by a malicious behavior which can be classified into two categories: 1) Direct attacks, 2) Indirect attacks. Ratha *et al.* define a comprehensive security and privacy framework that captures eight possible adversarial attack scenarios to a biometric recognition system [4] as depicted in Figure 7.1: 1- Spoofing&Mimicry attack, 2- Replay attack, 3- Attack against the feature extractor, 4- Tampering the communication channel between the feature extractor and the matcher, 5- Attack against the matcher, 6- Attack against the database, 7-Tampering the communication channel between the database and the matcher, 8- Override Response.

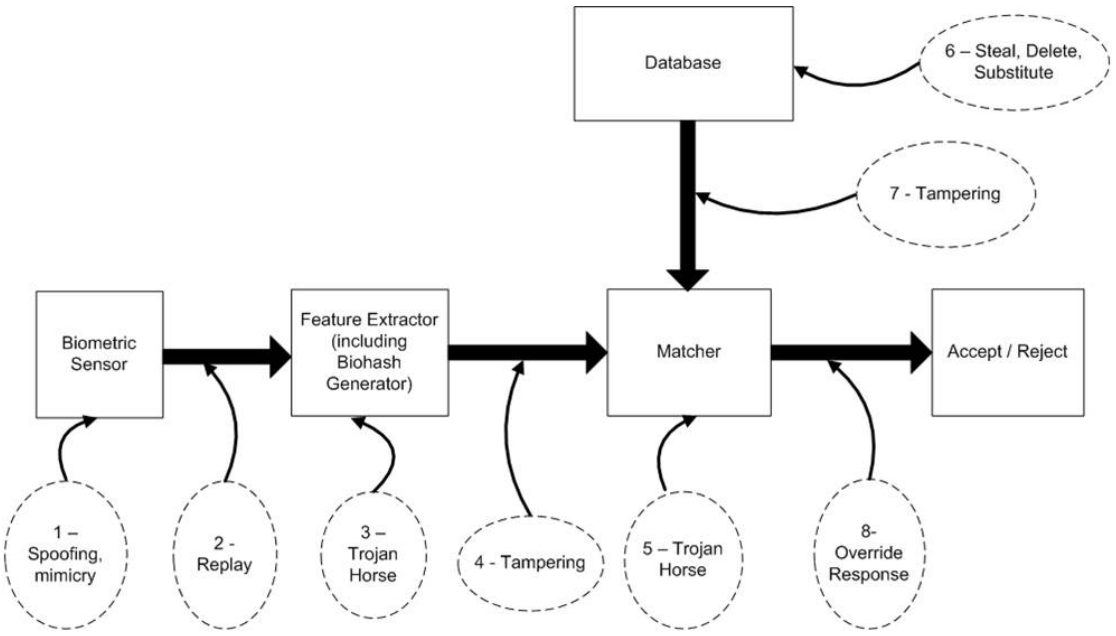

FIGURE 7.1: Possible attack points to a biometric recognition system (adapted from [4]).

7.2.2.1 Direct Attacks

Only spoofing (physiological biometric traits) & mimicry (behavioral biometric traits) attacks can be classified under direct attacks. These kinds of attacks are carried out by presenting a fake biometric template to the sensor, thus the adversary aims to gain access to the system by impersonating an authentic user [51, 52, 110]. The definitions of these attacks are given in the below.

Attack 1.a - Spoofing attack: This attack corresponds to the first attack type in Figure 7.1. In this attack, a fake biometric trait (i.e., face, fingerprint, face) or an artifact (i.e. gummy finger, high quality face image) is introduced to the sensor of a biometric recognition system in order to gain access to the system as a legitimate user [111, 112]. This attack is directly performed on the biometric sensor and the adversary does not need to know anything about the system (i.e. matching algorithm, feature extraction method).

Attack 1.b - Mimicry attack: This attack also corresponds to the first attack type in Figure 7.1. In this attack, a fake behavioral trait (e.g., signature, voice) is introduced to the sensor of a biometric recognition system in order to gain access to the system as a legitimate user [113]. In this type of attack, the adversary tries to break the system by imitating a legitimate user. This attack is directly performed on the biometric sensor and the adversary does not need to know anything about the system (i.e. matching algorithm, feature extraction method).

7.2.2.2 Indirect Attacks

Indirect attacks, on the other hand, include all the remaining seven attacks depicted in Figure 7.1. A Trojan horse might be used in order to carry out the third attack (attack against the feature extractor) and the fifth attack (attack against the matcher). In these cases, an adversary replaces the feature extractor or the matcher respectively, and outputs a feature vector or matching score (also final decision) different from the original. The sixth attack (attack against the database) targets the system database where an adversary aims to manipulate biometric templates (steal, delete, substitute, change, or add) in order to gain access to the system. The second attack type (replay attack), the fourth attack type (tampering the communication channel between the feature extractor and the matcher), the seventh attack type (tampering the communication channel between the database and the matcher) and the eight attack type (override response) focus on exploiting possible vulnerabilities in the communication channels of a biometric recognition system in order to extract, add or change the information. One of the most common indirect attack is the hill-climbing attack which can be performed by using the output of the matcher [30–33]. The definitions of aforementioned attacks are given in the below.

Attack 2 - Replay attack between the biometric sensor and the feature extractor: This attack corresponds to the second attack type in Figure 7.1. It uses weaknesses in the communication channel between the biometric sensor and the feature extractor in order to gain access to the system. In this attack, an adversary eavesdrops on the communication channel between the biometric sensor and the feature extractor and tries to obtain the biometric template of a user. When message interchange is over, the adversary connects to the feature extractor and sends the obtained biometric template of the user to the feature extractor in order to gain access to the system as a legitimate user.

Attack 3 - Attack against the feature extractor: This attack corresponds to the third attack type in Figure 7.1. It can be performed by using a Trojan horse that replaces the feature extractor and outputs a feature vector different than the original.

Attack 4 - Tampering the communication channel between the feature extractor and the matcher: This attack corresponds to the fourth attack type in Figure 7.1. In this attack, an adversary aims either to obtain information from the communication channel or to change information in the communication channel.

Attack 5 - Attack against the matcher: This attack corresponds to the fifth attack type in Figure 7.1. It can be performed by using a Trojan horse that replaces the matcher and outputs a matching score different than the original.

Attack 6 - Attacks against the database: This attack corresponds to the sixth attack type in Figure 7.1. In this attack, an adversary aims to manipulate biometric templates (steal, delete, substitute, change, or add) stored in the database.

Attack 7 - Tampering the communication channel between the database and the matcher: This attack corresponds to the seventh attack type in Figure 7.1. In this attack, an adversary aims either to get information from the communication channel or to change the information in the communication channel by manipulating the biometric data.

Attack 8 - Override response: This attack corresponds to the eight attack type in Figure 7.1. In a biometric authentication system, the verifier sends its decision (either Accept or Reject) to the user after making some computations. In this attack, an adversary aims to change the response of the verifier in order to gain access to the system.

Attack 9 - Hill-climbing attack: In this attack, an adversary iteratively updates a synthetically created biometric template using a specific modification scheme according to the score given by the matcher. It is assumed that the adversary can access the scores. When the score exceeds a fixed decision threshold, adversary gains access to the system.

7.3 Preliminaries

7.3.1 Threshold Homomorphic Cryptosystem

In this section, we briefly describe underlying cryptographic primitives of the protocols. Given a public key encryption scheme, let $m \in \mathcal{M}$ denote its message or plaintext space, $c \in C$ the ciphertext space, and $r \in \mathcal{R}$ its randomness. Let $c = \mathsf{Enc}_{pk}(m; r)$ depict an encryption of m under the public key pk where r is a random value. Let sk be its corresponding private key, which allows the holder to retrieve a message from a ciphertext. The decryption is done with the private key sk as $m = \mathsf{Dec}_{sk}(c)$.

In a (t, n)-threshold cryptosystem, the aim is to distribute the knowledge of a private key among parties $P_1, \ldots, P_n$ such that at least t of these parties are required for successful decryption. In

TABLE 7.1: Comparison between the THRIVE system and the existing solutions

	THRIVE	Salting Based Schemes	Non-Invertible Transform Based Schemes	Key Binding Schemes	Key Generation Schemes	[83]	[87]	[84]	[54, 86]
Irreversibility	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Cancelability	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Diversity	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Helper Data Usage	No	No	No	Yes	Yes	Yes	No	No	No
Homomorphic Encryption	Yes	No	No	No	No	No	Yes	Yes	Yes
Threshold Homomorphic Encryption	Yes	No	No	No	No	No	No	No	No
Malicious Attack Model	Yes	No	No	No	No	No	No	No	No

other words, each party holds a 'share' of the private key, the overall private key can collaboratively be reconstructed to let them recover the message in a given ciphertext. On the other hand, there is a public key that is used as in a regular public key scheme to perform decryption. More formally, let $P_1, \ldots, P_n$ be the participants. We define a (t, n)-threshold encryption scheme with three phases as follows:

- In the ***key generation*** phase, each participant P_i receives a pair (pk_i, sk_i), where pk_i and sk_i are the *shares* of the public and secret key, respectively. Then, the overall public key pk is constructed by collaboratively *combining* the shares. Finally pk is broadcast to allow anyone to encrypt messages in $\mathcal{M}$. The shares of this public key are also broadcast which allow all parties to check the correctness of the decryption process.

- The ***encryption*** phase is done as in any public key encryption cryptosystem. If $m \in \mathcal{M}$ is the message, a (secret) random value r from $\mathcal{R}$ is chosen and $c = \mathsf{Enc}_{pk}(m; r)$ is broadcast under a public key pk.

- In the ***threshold decryption*** phase, given that t (or more) participants agree to decrypt a ciphertext c, they follow two steps. First, each participant produces a decryption share

by performing $S_{i_j} = \mathsf{Dec}_{sk_{i_j}}(c)$, $j = 1, \ldots, t$. After broadcasting S_{i_j}, they all can apply a reconstruction function $\mathcal{Z}$ on these shares so that they can recover the original message by performing $m = \mathcal{Z}(S_{i_1}, \ldots, S_{i_t})$ where $P_{i_1}, \ldots, P_{i_t}$ represent the group of t participants willing to recover m.

In case of a (t, n)-threshold scheme, the additional requirement is that if less than t parties gather their correct shares of the decryption of a given ciphertext, they will get no information whatsoever about the plaintext. In the proposed system, we use the $(2, 2)$-threshold cryptosystem between the claimer (the user) and the verifier where both players must cooperate in order to decrypt.

A public key encryption scheme is said to be additively homomorphic if given $c_1 = \mathsf{Enc}(m_1; r_1)$ and $c_2 = \mathsf{Enc}(m_2; r_2)$ it follows that $c_1 c_2 = \mathsf{Enc}(m_1 + m_2; r_3)$. As a consequence, it is also true that $\mathsf{Enc}(m; r)^s$ is equal to $\mathsf{Enc}(ms; rs)$ for a known integer s. Another consequence of these properties is the re-randomization of encryption, by observing that $\mathsf{Enc}(m; r)\mathsf{Enc}(0; r')$ is a new encryption whose plaintext is again m (and its randomness is r'').

There are various versions of threshold homomorphic cryptosystems. The most widely used are ElGamal [114] or Paillier [115] cryptosystems. Threshold homomorphic ElGamal has the drawback of only allowing decryption of values belonging to a relatively small set, for which it is feasible to compute discrete logs. On the other hand, Paillier does not have this problem and allows decryption of encrypted values in an arbitrarily large set (e.g., 1024-bit integers). However, the distributed key generation protocol for threshold Paillier is very expensive compared to that for threshold ElGamal. In the proposed protocol, we consider a variant of the threshold decryption protocol, the so-called private threshold decryption [116]. Here, the requirement is that one of the t parties will be the only party who will recover the secret. This is easily achieved: all $t - 1$ other parties follow the protocol, and broadcast their shares (along with the proofs of correctness). The party who will learn the plaintext proceeds with the decryption process privately, collects all decryption shares from the $t - 1$ other parties, and privately reconstructs the message. Note that the remaining parties will not get any information about this message.

7.3.2 The Paillier Encryption System

The Paillier cryptosystem, which is invented by Pascal Paillier in 1999, is a probabilistic asymmetric algorithm for public key cryptography [115]. In this part, we briefly describe the Paillier

encryption algorithm. In the key generation phase; an RSA common modulus $n = pq$ with p and q large primes is generated and the Carmichael function of $n : \lambda = lcm(p-1, q-1)$ is computed. Then, a generator g of $Z^*_{n^2}$ such that $g = 1(mod\ n)$ is computed. Next, the public key, (n, g) is published while λ forms the secret key. Finally, the encryption of a message $m \in Z_n$ is computed as:

$$c = g^m r^n\ (mod\ n^2), \tag{7.1}$$

where $r \in_R Z^*_n$.

Moreover, the decryption is performed as follows:

$$m = \frac{L(c^\lambda (mod\ n^2))}{L(g^\lambda (mod\ n^2))} (mod\ n), \tag{7.2}$$

where, for convenience, we simply define $L(x) = (x-1)/n$.

7.3.3 Digital Signatures

A digital signature is a cryptographic primitive which is used for proving the authenticity of its digital content. The digital signatures are used to associate the user's identity to the electronic documents. They should be easy to produce, check and difficult to forge. They are generated by using the private key and verified by using the public key. Thus, only the user can perform digital signature operation whereas everybody can verify it. Eventually, the digital signature schemes have two main stages:

- Signature creation,
- Signature verification.

In the literature, there are two widely used digital signature schemes: 1) RSA digital signature scheme, 2) Digital Signature Algorithm (DSA) [117]. Here we describe the RSA digital signature scheme.

Signature creation stage: A signer follows the below steps for performing signature operation:

1. She chooses secret odd primes p, q and computes $n = pq$.

2. She chooses e_A with $gcd\,(e_A, \Phi\,(n)) = 1$ where $gcd(.)$ denotes the greatest common divisor function of its inputs and $\Phi\,(n)$ denotes Euler's phi function.

3. She computes $d_A = e_A^{-1} \mod \Phi\,(n)$. Therefore, she computes her public (e_A, n) and private keys (d_A, p, q).

4. She signs the message m as in the following.

$$y \equiv m^{d_A} \mod n. \tag{7.3}$$

5. She sends the signed message (y, m) to the verifier.

Signature verification stage: A verifier receives (y, m) and downloads the signer's public keys (e_A, n). Then, he computes

$$z \equiv y^{e_A} \mod n. \tag{7.4}$$

The signature is valid iff $m \equiv z$.

7.3.4 Biometric Verification Scheme

Biometric verification schemes perform an automatic verification of a user based on her specific biometric data (e.g., face, fingerprint, iris). There are two main stages in these schemes: 1) Enrollment stage, and 2) Authentication stage. The user is enrolled to the system at the enrollment stage. Then, the user again provides her biometric data to the system at the authentication stage in order to prove her identity. Any biometric scheme, which provides binary fixed size templates or whose templates can be binarized, can work with the proposed threshold homomorphic cryptosystem. The THRIVE system can work with any biometric feature extraction method which produces fixed size vectors as templates and perform verification with distance calculations between the enrolled and the provided template at the authentication stage. When the output of a biometric feature extraction method is not binary, locality sensitive hashing can be used in order to binarize the feature vector [118]. After binarization, the binary templates can successfully be used with the proposed system. In this chapter, we use biohashing as an example algorithm

for extracting binary biometric templates. Although biohashing has its own security and privacy preservation mechanism, we do not rely on these for the security or the privacy preservation of the proposed system. Thus it can be replaced with any other binary feature extraction method. We call the fixed size binary biometric templates as "biohash" in this chapter although we are not limited to "biohashing" method for obtaining them.

Biohashing schemes are simple yet powerful biometric template protection methods [16, 18–21]. Biohash is a binary and pseudo-random representation of a biometric template. Biohashing schemes use two inputs: 1) Biometric template, 2) User's secret key. A biometric feature vector is transformed into a lower dimension sub-space using a pseudo-random set of orthogonal 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. In an ideal case, the distance between the biohashes belonging to the 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 to achieve higher recognition rates.

In this part, we describe the random projection (RP) based biohashing scheme proposed by Ngo *et al.* [2]. In a RP based biohashing scheme, there are three main phases: 1) Feature extraction, 2) Random projection, 3) Quantization. These steps for the face biometric are explained below.

7.3.4.1 Feature Extraction

At this phase, face images in the training set, which are collected during the enrollment stage, are used. The set has training face images belonging to the registered users, $\mathbf{I}_{i,j} \in \mathbb{R}^{m \times n}$ where $i = 1, \ldots, K$ and K denotes number of users, $j = 1, \ldots, L$ and L denotes number of training images per user. The faces images are lexicographically re-ordered and the training face vectors, $x_{i,j} \in \mathbb{R}^{(mn) \times 1}$, are obtained. Then, Principle Component Analysis (PCA) [34] is applied to the face images in the training set for feature extraction as follows:

$$\boldsymbol{y}_{i,j} = \mathbf{A}(\boldsymbol{x}_{i,j} - \boldsymbol{w}), \tag{7.5}$$

where $\mathbf{A} \in \mathbb{R}^{k \times (mn)}$ is the PCA matrix trained by the face images in the training set, $\boldsymbol{w}$ is the mean face vector, and $\boldsymbol{y}_{i,j} \in \mathbb{R}^{k \times 1}$ is vector containing PCA coefficients belonging to the j^{th} training image of the i^{th} user.

7.3.4.2 Random Projection

At this phase, a RP matrix, $\mathbf{R} \in \mathbb{R}^{\ell \times k}$, is generated to reduce the dimension of the PCA coefficient vectors. The RP matrix elements are independent and identically distributed (*i.i.d*) and generated from a Gauss distribution with zero mean and unit variance by using a Random Number Generator (RNG) with a seed derived from the user's secret key. The Gram-Schmidt (GS) procedure is applied to obtain an orthonormal projection matrix $\mathbf{R}_{GS} \in \mathbb{R}^{\ell \times k}$ to have more distinct projections. Finally, PCA coefficients are projected onto a lower ℓ-dimensional subspace as follows:

$$\boldsymbol{z}_{i,j} = \mathbf{R}_{GS} \boldsymbol{y}_{i,j}, \tag{7.6}$$

where $\boldsymbol{z}_{i,j} \in \mathbb{R}^{\ell \times 1}$ is an intermediate biohash vector belonging to the j^{th} training image of the i^{th} user.

7.3.4.3 Quantization

At this phase, the intermediate biohash vector $\boldsymbol{z}_{i,j}$ elements are binarized with respect to a threshold as follows:

$$\lambda_{i,j}(k) = \begin{cases} 1 & \text{if} \quad \boldsymbol{z}_{i,j}(k) \geq \beta, \\ 0 & \text{Otherwise}, \end{cases} \tag{7.7}$$

where $\lambda_{i,j} \in \{0,1\}^{\ell}$ denotes biohash vector of the j^{th} training image of the i^{th} user and β denotes the mean value of the intermediate biohash vector $\boldsymbol{z}_{i,j}$.

The computed binary biohashes are stored in the database in the enrollment stage for verification purpose during the authentication stage. The user is authenticated when the Hamming distance between $\boldsymbol{B}_{enroll}$ (which denotes the biohash of the user generated at the enrollment stage) and $\boldsymbol{B}_{auth}$ (which denotes the biohash of the user generated at the authentication stage) is below a pre-determined distance threshold μ as follows:

$$\sum_{k=1}^{n} \boldsymbol{B}_{enroll}(k) \oplus \boldsymbol{B}_{auth}(k) \leq \mu, \tag{7.8}$$

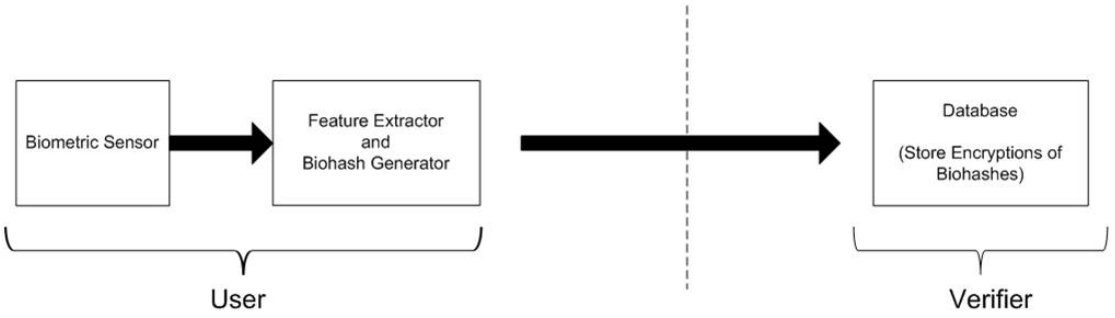

FIGURE 7.2: Illustration of the THRIVE enrollment stage: the user has control over the biometric sensor, the feature extractor and the biohash generator whereas the verifier has control over the database.

where $\oplus$ denotes the binary XOR (exclusive OR) operator. Therefore, the verifier decides whether the user is legitimate or not using the pre-defined distance threshold.

7.4 The Proposed Biometric Authentication System

In this section, the proposed biometric authentication system is introduced. In the proposed system, there are two major roles: 1. *User* (U) and 2. *Verifier* (V). The user has control of the biometric sensor, the feature extractor, and the biohash generator whereas the verifier has control of the database and the matcher. We assume that there is a trusted third party (TTP) which initially sets up the system public/private keys.

The TTP distributes the keys in the proposed THRIVE system. There are public-private key pairs $(pk_i, (sk_i^1, sk_i^2))$ which are shared between the user and the verifier. pk_i is the public key of the i^{th} user, U_i, and both the user and the verifier have it. Recall that, when an enrollment biometric template is encrypted by pk_i, this can solely be decrypted using the private key shares of the user (sk_i^1) and the verifier (sk_i^2) collaboratively since the proposed system is based on the $(2, 2)$-threshold homomorphic cryptosystem. Here, sk_i^1 is the private key share of the i^{th} user, U_i, and sk_i^2 is the private key share of the verifier. Besides, there is a public-private key pair (pk_{U_i}, sk_{U_i}) which belongs to the i^{th} user, U_i, where pk_{U_i} is the public key and sk_{U_i} is its associated private key in order to perform the signature operation. The verifier also has the public key pk_{U_i} of the i^{th} user, U_i.

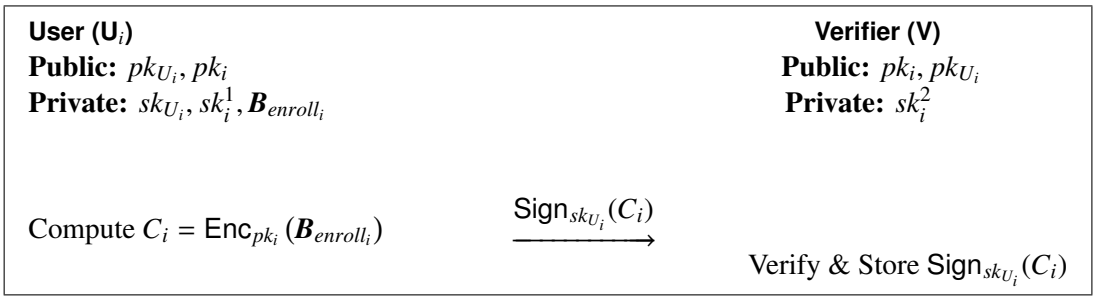

FIGURE 7.3: The THRIVE Enrollment Protocol

7.4.1 Enrollment Stage

Access rights of the verifier and the user to the biometric authentication system at the enrollment stage are illustrated in Figure 7.2. At this stage, the i^{th} user, U_i, has control over the biometric sensor, the feature extractor, and the biohash generator whereas the verifier has control over the database. It is worth mentioning that biometric sensor authentication must be achieved in the proposed system before executing the enrollment protocol to prevent unauthorized sensors to be used as clients in the system by malicious users. However this is not explicitly indicated in the protocol in order not to clutter the chapter. The proposed enrollment protocol is illustrated in Figure 7.3 and steps of the proposed enrollment protocol are introduced as follows:

1. **Step 1**: The i^{th} user, U_i, computes her biohash, $\boldsymbol{B}_{enroll_i}$. Next, the user encrypts her biohash, $C_i = \mathsf{Enc}_{pk_i}(\boldsymbol{B}_{enroll_i})$, by using the public key pk_i. Then, the user signs her encrypted biohash, $\mathsf{Sign}_{sk_{U_i}}(C_i)$, and sends it to the verifier.

2. **Step 2**: The verifier verifies $\mathsf{Sign}_{sk_{U_i}}(C_i)$ of U_i by using pk_{U_i} and stores it in the database. These data will be used for verification at the authentication stage.

Recall that the proposed enrollment protocol uses the $(2, 2)$-threshold homomorphic cryptosystem. Namely, both the user and the verifier have to cooperate in order to decrypt a ciphertext. Furthermore, the signature ensures that the data stored in the database are generated by a legitimate user.

Lemma 1. *Biohashes are not revealed at the enrollment stage.*

Proof. (Sketch) At the enrollment stage, user first encrypts her biohash and then signs it. After these computations, the user sends her encrypted and signed biohash ($\mathsf{Sign}_{sk_{U_i}}(C_i)$) to the verifier. Since the user's biohash is not sent in plain form, biohashes are not revealed at the enrollment stage.

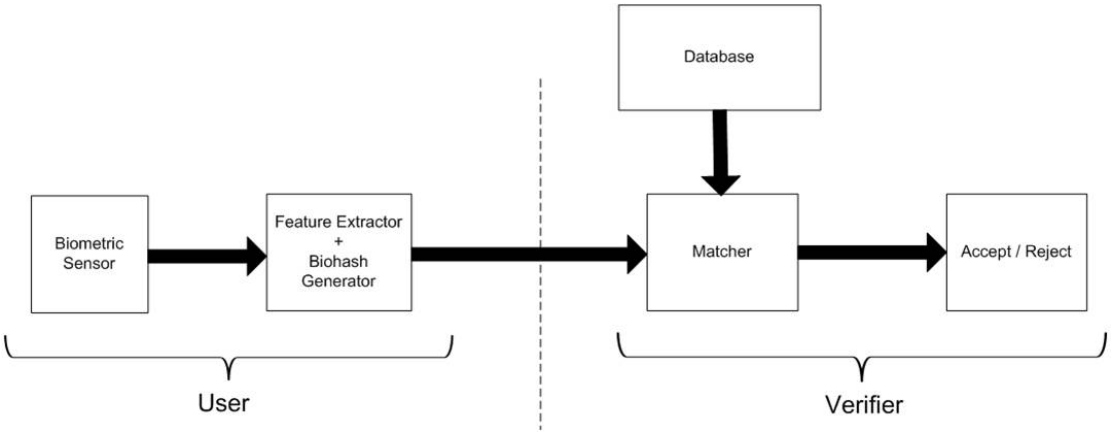

FIGURE 7.4: Illustration of the THRIVE authentication stage: the user has control over the biometric sensor, the feature extractor and the biohash generator whereas the verifier has control over the database, the matcher and the decision maker.

Lemma 2. *An adversary cannot register as a legitimate user at the enrollment stage.*

Proof. (Sketch) At the enrollment stage, user encrypts her biohash by using the public key pk_i and then signs her encrypted biohash by using her private key sk_{U_i}. Thus, the user sends encrypted and signed biohash ($\mathsf{Sign}_{sk_{U_i}}(C_i)$) to the verifier. The verifier knows pk_{U_i} of each genuine user. Since verifier verifies the signature of the user, an adversary cannot register himself as a genuine user without having the private key of the user sk_{U_i} for computing $\mathsf{Sign}_{sk_{U_i}}(C_i)$.

7.4.2 Authentication Stage

Access rights of the verifier and the user to the biometric authentication system at the authentication stage are illustrated in Figure 7.4. At this stage, the i^{th} user, U_i, has control over the biometric sensor, the feature extractor, and the biohash generator whereas the verifier has control over the database, the matcher and the decision maker. The user U_i tries to prove herself to the verifier by executing the proposed authentication protocol shown in Figure 7.5. Similar to the enrollment case, the biometric sensor must be authorized by the system before the authentication protocol is carried out. Steps of the proposed authentication protocol are as introduced as follows:

1. **Step 1**: The i^{th} user wants to verify her identity to the verifier by using her biohash and sends connection request to the verifier. Then, the i^{th} user, U_i, computes her biohash $\boldsymbol{B}_{auth_i}$. Note that the user cannot produce exactly the same biometric template at each attempt and this results in different biohashes computed by the same user. Therefore, $\boldsymbol{B}_{enroll_i}$

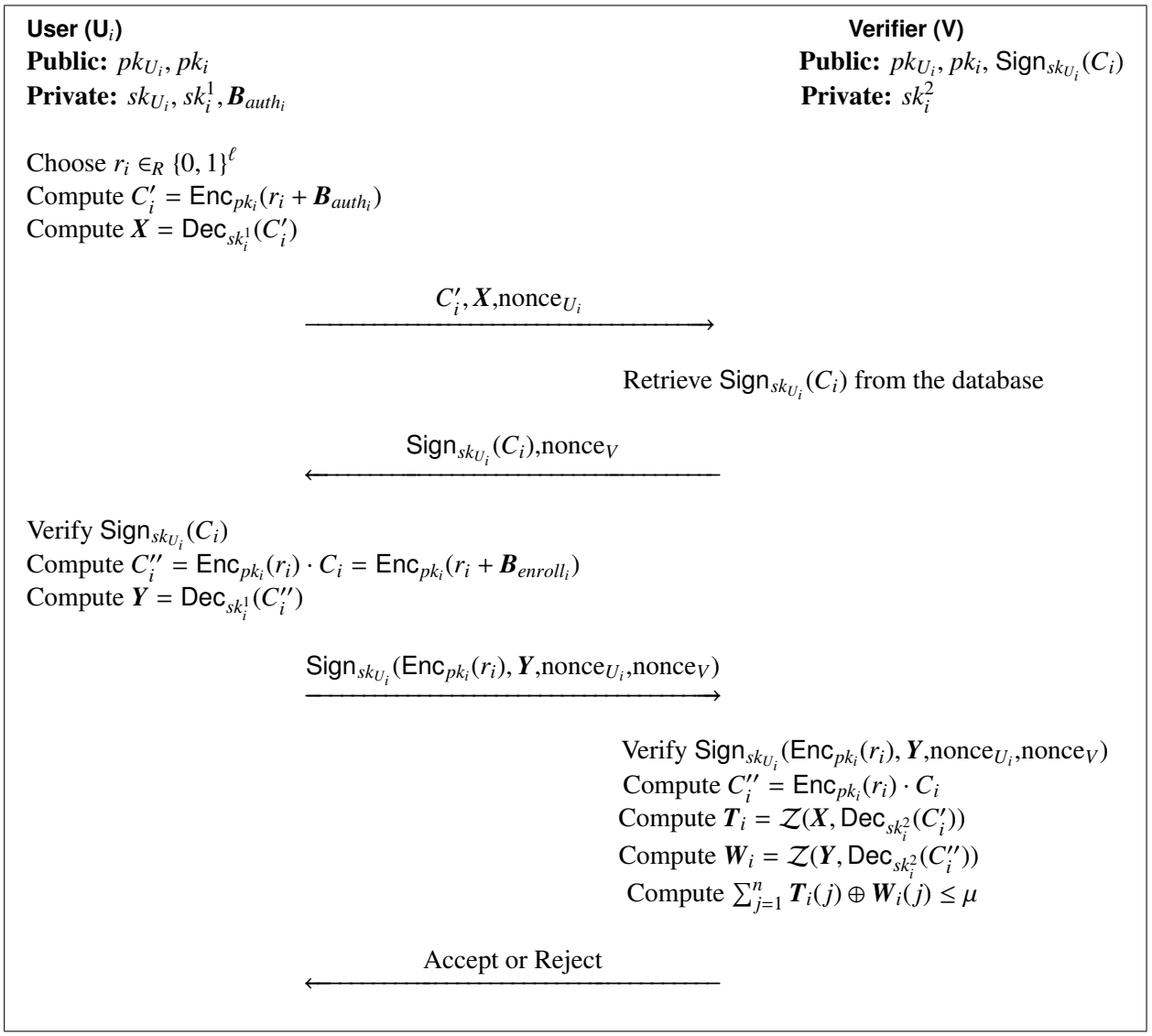

FIGURE 7.5: The THRIVE Authentication Protocol

and $\boldsymbol{B}_{auth_i}$ are different although they are assumed to be generated by the same user at different sessions (enrollment and authentication). First of all, the user chooses a random vector $\boldsymbol{r} \in_R \{0,1\}^\ell$ where ℓ is the length of the biohash vector and $\in_R$ denotes that the variable is chosen uniformly randomly. She computes $C_i' = \mathsf{Enc}_{pk_i}(r_i + \boldsymbol{B}_{auth_i})$ by using the public key pk_i. Then, she performs partial decryption over C_i', which is denoted by $\boldsymbol{X} = \mathsf{Dec}_{sk_i^1}(C_i')$, by using her private key share sk_i^1. Then, the i^{th} user generates a nonce called nonce_{U_i} which is an arbitrary number which is used only once in a cryptographic communication. nonce_{U_i} contains information about user id, session id and time stamp. Finally, the user sends C_i', $\boldsymbol{X}$, nonce_{U_i} to the verifier.

2. **Step 2**: The verifier retrieves $\mathsf{Sign}_{sk_{U_i}}(C_i)$ from the database where $C_i = \mathsf{Enc}_{pk_i}(\boldsymbol{B}_{enroll_i})$. Then, it generates a nonce called nonce_V which contains information about the verifier, session id and time stamp. Finally, it sends $\mathsf{Sign}_{sk_{U_i}}(C_i)$,$\text{nonce}_V$ to the user.

3. **Step 3**: The user verifies $\mathsf{Sign}_{sk_{U_i}}(C_i)$ by using public key pk_{U_i}. She computes $C_i'' = \mathsf{Enc}_{pk_i}(r_i + \boldsymbol{B}_{enroll_i})$. Thanks to the homomorphic encryption scheme properties. Note that $C_i'' = \mathsf{Enc}_{pk_i}(r_i + \boldsymbol{B}_{enroll_i}) = \mathsf{Enc}_{pk_i}(r_i) \cdot C_i$ due to the homomorphic encryption. Then, she performs partial decryption over C_i'', which is denoted by $\boldsymbol{Y} = \mathsf{Dec}_{sk_i^1}(C_i'')$, by using her private key share sk_i^1. Finally, she sends $\mathsf{Sign}_{sk_{U_i}}(\mathsf{Enc}_{pk_i}(r_i), \boldsymbol{Y}, \text{nonce}_{U_i}, \text{nonce}_V)$ to the verifier.

4. **Step 4**: The verifier verifies the signature, $\mathsf{Sign}_{sk_{U_i}}(\mathsf{Enc}_{pk_i}(r_i), \boldsymbol{Y}, \text{nonce}_{U_i}, \text{nonce}_V)$, by using the public key pk_{U_i}. Then, it computes $C_i'' = \mathsf{Enc}_{pk_i}(r_i) \cdot C_i$ since it does not trust the user. Note that the verifier also has to compute C_i'' in order to verify that C_i'' is computed correctly by the user. Next, it performs full decryption by using reconstruction function, $\mathcal{Z}(.)$, whose inputs are $\boldsymbol{X}$ and $\mathsf{Dec}_{sk_i^2}(C_i')$ and computes $T_i = \boldsymbol{r}_i + \boldsymbol{B}_{enroll_i}$. Similarly, it performs full decryption by using reconstruction function, $\mathcal{Z}(.)$, whose inputs are $\boldsymbol{Y}$ and $\mathsf{Dec}_{sk_i^2}(C_i'')$ and computes $W_i = \boldsymbol{r}_i + \boldsymbol{B}_{auth_i}$. Finally, the Hamming distance [95] between $\boldsymbol{T}_i = \boldsymbol{r}_i + \boldsymbol{B}_{enroll_i}$ and $\boldsymbol{W}_i = \boldsymbol{r}_i + \boldsymbol{B}_{auth_i}$ is computed and and compared to a threshold as follows:

$$\sum_{j=1}^{n} \boldsymbol{T}_i(j) \oplus \boldsymbol{W}_i(j) \leq \mu, \tag{7.9}$$

where $\boldsymbol{T}_i(j)$ and $\boldsymbol{W}_i(j)$ is the j^{th} bit of $\boldsymbol{T}_i$ and $\boldsymbol{W}_i$ respectively, μ is a pre-defined distance threshold. Therefore, the verifier decides whether the user is authentic with respect to the pre-defined distance threshold. Note that the Hamming distance between $\boldsymbol{T}_i = \boldsymbol{r}_i + \boldsymbol{B}_{enroll_i}$ and $\boldsymbol{W}_i = \boldsymbol{r}_i + \boldsymbol{B}_{auth_i}$ is equal to the Hamming distance between $\boldsymbol{B}_{enroll_i}$ and $\boldsymbol{B}_{auth_i}$.

5. **Step 5**: Finally, the verifier sends its decision (either Accept or Reject) to the user. However, the user may get dummy output if there is an error or an attack (i.e., override response attack) in the communication channel. The proposed system can easily be updated to cope with such an attack, for instance, by allowing the verifier to sign its decision including the nonces generated during the authentication session (i.e. either Sign(Accept,

nonce_{U_i}, nonce_V)or Sign (Reject, nonce_{U_i}, nonce_V) and then send it to the user. In this way, authenticity, integrity and origin of the data can easily be verified. Besides, signing the nonces (nonce_{U_i} and nonce_V) also make the communication unique and avoids replay attacks.

Lemma 3. *Biohashes are not revealed at the authentication stage.*

Proof. Authentication is performed in a randomized domain. In other words, the authentication is determined by comparing $\boldsymbol{T}_i$ and $\boldsymbol{W}_i$. An adversary can only obtain $\boldsymbol{T}_i$ and $\boldsymbol{W}_i$ which are revealed at the authentication stage. Recall that these are randomized biohashes. Thus, from the adversary's perspective, there are three unknowns ($\boldsymbol{r}_i$, $\boldsymbol{B}_{enroll_i}$ and $\boldsymbol{B}_{auth_i}$) and two equations which are shown in the below.

$$\boldsymbol{T}_i = \boldsymbol{r}_i + \boldsymbol{B}_{enroll_i}, \tag{7.10}$$

$$\boldsymbol{W}_i = \boldsymbol{r}_i + \boldsymbol{B}_{auth_i}, \tag{7.11}$$

where r_i is the random number generated by the i^{th} user. Since this is a system of linear equations with fewer equations than unknowns, the system has infinitely many solutions. Consequently, it is impossible for the adversary to obtain a legitimate user's biohash by using $\boldsymbol{T}_i$ and $\boldsymbol{W}_i$ which are revealed at the authentication stage. As a result, the proposed biometric authentication system ensures security and privacy.

7.5 Security and Privacy Analysis

In this section, the security and privacy preservation capability of the THRIVE system is analyzed. Further details are given below.

7.5.1 Security and privacy arguments against possible attacks

In addition to the malicious party attacks, Ratha *et al.* classify adversary attacks against biometric systems into two groups [4] in the literature: 1) Direct attacks, 2) Indirect attacks. They define eight main attack points to a biometric system as illustrated in Figure 7.1. These attacks

are introduced in Section II - Attacks on Biometric Systems. In these attacks, there are two main threats:

1. the adversary can threaten the security by gaining access to the proposed system,
2. the adversary can threaten the privacy by obtaining the biometric data of a user or the users.

There are some other attack types in the literature e.g. denial-of-service (DoS) attack [119]. DoS attack is an adversary attack which aims to make the system unavailable to its legitimate users. Note that, when an adversary just aims to prevent a legitimate user from gaining access to the proposed system, he can succeed in such DoS attacks. The proposed system does not provide protection against DoS attacks and these type of attacks are out of scope for this chapter. The proposed system aims to improve the security against illegal authentication attempts. Besides, the proposed system enhances privacy of the biometric data of the users.

In this section, informal security and privacy analysis of the proposed biometric authentication system is made against the adversary attacks (including hill climbing attack which is a special case of the indirect attacks) which are classified as follows:

1. Direct Attacks
   (a) Attack 1 - Spoofing & Mimicry attacks
2. Indirect Attacks
   (a) Attack 2 - Replay attack
   (b) Attack 3 - Attack against the feature extractor
   (c) Attack 4 - Tampering the communication channel between the feature extractor and the matcher
   (d) Attack 5 - Attack against the matcher
   (e) Attack 6 - Attack against the database
   (f) Attack 7 - Tampering the communication channel between the database and the matcher
   (g) Attack 8 - Override response
   (h) Attack 9 - Hill-climbing attack

1. Protection against Attack 1 - Spoofing & Mimicry attack: An adversary can neither gain access to the proposed system nor obtain biometric template of a user by performing a spoofing & mimicry attack. Let us assume that an adversary introduces a fake biometric data to the sensor in order to impersonate a real user. Even if the fake biometric data is very similar to the original biometric data of the user, the adversary cannot gain access to the proposed system since he does not know the private keys of the user (i.e. the user's private key share (sk_i^1) and the user's private key sk_{U_i} for signature) which are also required at the authentication stage. In other words, the proposed system is a two-factor authentication system where a claimer must provide biometric data and her private key simultaneously at each authentication attempt.

On the other hand, let us assume that the adversary obtains private keys of the user. In this case, security of the proposed system depends on the secrecy of the biometric data. As soon as the adversary does not provide similar biometric data, he cannot gain access to the proposed system. Note that the adversary cannot obtain the original biometric data of the user even if he obtains private keys of the user since the (2,2)-threshold homomorphic cryptosystem is used. Therefore, the adversary cannot perform decryption over $C_i = \mathsf{Enc}_{pk_i}(\boldsymbol{B}_{enroll_i})$ since he does not know private key share of the user (i.e. sk_i^1). Consequently, the proposed system also preserves the privacy of the users.

2. Protection against Attack 2 - Replay Attack: An adversary cannot gain access to the proposed system by performing a replay attack. Let us assume that an adversary eavesdrops on the communication channel between the biometric sensor and the feature extractor. Then, he can only obtain the biometric template of a legitimate user. The adversary, however, cannot gain access to the proposed system since he does not know private keys of the user (i.e. the user's private key share (sk_i^1) and the user's private key sk_{U_i} for signature) which are also required at the authentication stage. Recall that the proposed system is a two-factor authentication system and the system security depends on the secrecy of private keys of the user in this case.

3. Protection against Attack 3 - Attack against the feature extractor: An adversary cannot gain access to the proposed system by performing an attack against the feature extractor. Let us assume that an adversary targets the feature extractor and uses trojan horse in order to obtain the biometric feature vector of the legitimate user. Even if the adversary can obtain the feature vector, he cannot gain access to the proposed system since he does not know the private keys of the user (e.g. the user's private key share (sk_i^1) and the user's private key sk_{U_i} for signature).

Recall that the proposed system is a two-factor authentication system where a claimer must provide biometric data together with her private key at each authentication attempt. In this case, the system security depends on the secrecy of the private keys of the user.

4. Protection against Attack 4 - Tampering the communication channel between the feature extractor and the matcher: An adversary can neither gain access to the proposed system nor can obtain biometric template of a user by performing an attack on the communication channel between the feature extractor and the matcher. Recall that the communication channel between the feature extractor and the matcher is actually the communication channel between the user and the verifier as depicted in Figure 7.2 and Figure 7.4. Thus, it may suffer from several attacks:

i) Replay attack

ii) Man-in-the-middle attack

iii) Ciphertext-only attack

iv) Known plaintext attack

i)Replay attack in the communication channel between the feature extractor and the matcher: Let us assume that an adversary eavesdrops on the communication channel between the user and the verifier and records the exchanged messages in the proposed authentication protocol. In other words, let us assume that the adversary wants to perform a replay attack in this communication channel. In this case, the adversary aims either to use these data for deceiving the verifier and gaining access to the proposed system or to obtain the biometric template of a user in order to threaten privacy.

The proposed authentication protocol is secure against the replay attack in the communication channel between the user and the verifier since

1. the user uses nonce_{U_i} at the first step of the proposed authentication protocol,
2. the verifier uses nonce_V at the second step of the proposed authentication protocol,
3. the user signs the whole communication, $\mathsf{Sign}_{sk_{U_i}}(\mathsf{Enc}_{pk_i}(r_i), \boldsymbol{Y}, \text{nonce}_{U_i}, \text{nonce}_V)$, during the authentication stage at the third step of the proposed authentication protocol.

Since the nonce is different for each authentication request, this makes the communication between the user and the verifier unique for each authentication session. Moreover, the adversary cannot perform sign operation since he does not know the private key of the user sk_{U_i} and he must perform the third step of the authentication protocol for the new nonce introduced by the verifier. Therefore, the adversary cannot gain access to the proposed system by performing replay attack in the communication channel between the feature extractor and the matcher. On the other hand, the proposed authentication protocol preserves privacy of the user in case of the replay attack in the communication channel between the user and the verifier since only encrypted biometric templates can be obtained from the communication channel. Even if the adversary obtains the private keys of the user, he cannot decrypt and compromise the biometric template of the user since the $(2, 2)$ threshold homomorphic cryptosystem is used. In other words, the adversary also needs private key share of the verifier in order to perform a successful decryption.

ii)Man-in-the-middle attack attack in the communication channel between the feature extractor and the matcher: In the literature, man-in-the-middle attack is an active attack where an adversary makes independent connections with the users and relays messages between them. Let us assume that an adversary wants to perform man-in-the-middle attack in the communication channel between the user and the verifier. In this case, the adversary aims either to gain access to the proposed system or obtain the biometric template of the user. When the adversary eavesdrops on the communication channel between the user and the verifier, he can only obtain the below items.

1. C_i', $\boldsymbol{X}$, nonce_{U_i} from the first step of the proposed authentication protocol,
2. $\mathsf{Sign}_{sk_{U_i}}(C_i)$,$\text{nonce}_V$ from the second step of the proposed authentication protocol, and
3. $\mathsf{Sign}_{sk_{U_i}}(\mathsf{Enc}_{pk_i}(r_i), \boldsymbol{Y}, \text{nonce}_{U_i}, \text{nonce}_V)$ from the second step of the proposed authentication protocol.

However, all exchanged messages in the proposed authentication protocol are encrypted and randomized and the adversary cannot perform decryption to learn the content since he does not know the private keys of the user and the verifier. Thus, he cannot properly modify the content of the exchanged messages in the proposed authentication protocol in order to gain access to the proposed system as a legitimate user.

iii) Ciphertext-only attack in the communication channel between the feature extractor and the matcher: Another type of attack that can be performed in this communication channel is a ciphertext-only attack (COA). The COA is an attack model for cryptanalysis where an adversary is assumed to have access only to a set of ciphertexts. When corresponding plaintexts can be deduced or private key is obtained, the COA is successful. For our case, biohashes ($\boldsymbol{B}_{auth_i}$) are plaintexts and encryption of biohashes ($\mathsf{Enc}_{pk_i}(\boldsymbol{B}_{auth_i})$) are ciphertexts. Let us assume that the adversary obtains encryption of biohashes ($\mathsf{Enc}_{pk_i}(\boldsymbol{B}_{auth_i})$) and he wants to gain access to the proposed system. For this case, there are three options:

1. The adversary cannot deduce the plaintext (the user's biohash ($\boldsymbol{B}_{auth_i}$)) from the ciphertext (encryption of the user's biohash ($\mathsf{Enc}_{pk_i}(\boldsymbol{B}_{auth_i})$)) since he does not have the user's private key share (sk_i^1).

2. The adversary cannot use the verifier in order to decrypt $\mathsf{Enc}_{pk_i}(\boldsymbol{B}_{auth_i})$ since the verifier only sends $\mathsf{Sign}_{sk_{U_i}}(C_i)$ and nonce_V where $C_i = \mathsf{Enc}_{pk_i}(\boldsymbol{B}_{enroll_i})$ to the user at the second step of the proposed authentication protocol. Thus, the adversary does not obtain any useful information from the verifier since the verifier only sends the data stored in the database. Besides, it does not perform decryption and does not send these decrypted data back to the adversary. Furthermore, since the adversary also does not have the user's private key share (sk_i^1) and the verifier's private key share (sk_i^2), he cannot decrypt C_i.

3. The adversary cannot gain access to the proposed system even if he obtains biohash ($\boldsymbol{B}_{auth_i}$) and encrypted biohashes ($\mathsf{Enc}_{pk_i}(\boldsymbol{B}_{auth_i})$) of the user without having her private key share (sk_i^1).

Therefore, the adversary cannot gain access to the proposed system by using the COA in the communication channel between the user and the verifier.

iv)Known-plaintext attack in the communication channel between the feature extractor and the matcher: In addition to the above mentioned attacks, an adversary can perform known-plaintext attack (KPA) by tampering the communication channel between the user and the verifier. The KPA is an attack where the adversary has samples of plaintext and its encrypted version. The adversary aims to reveal further secret information such as private keys by using these information and gain access to the system. Let us assume that the adversary obtains biohashes ($\boldsymbol{B}_{auth_i}$) and encryption of biohashes ($\mathsf{Enc}_{pk_i}(\boldsymbol{B}_{auth_i})$) and he wants to gain access to

the proposed system as a legitimate user. He needs to execute the authentication protocol in order to gain access to the proposed system. In this manner, the adversary generates a random, $r_i \in_R \{0,1\}^{\ell}$. Thus, he can compute $C'_i = \mathsf{Enc}_{pk_i}(r_i + \boldsymbol{B}_{auth_i})$ by using the user's public key pk_i since public key of the user is known by all. However, he cannot perform $\boldsymbol{X} = \mathsf{Dec}_{sk_i^1}(C'_i)$ since he does not know the user's private key share (sk_i^1). Therefore, he cannot execute the proposed authentication protocol in order to enter the proposed system. Furthermore, the adversary cannot obtain further information about the user's private key share (sk_i^1) from the communication channel between the user and the verifier via the KPA since

1. all the exchanged messages are encrypted in the communication channel between the user and the verifier,
2. the verifier can only send the encrypted data stored in the database to the adversary.

These messages do not help the adversary for deducing the private key of the user. Therefore, the adversary cannot gain access to the proposed system by using the KPA in the communication channel between the user and the verifier.

5. Protection against Attack 5 - Attack against matcher: An adversary cannot obtain biometric template of a user even if he gains access to the proposed system by performing an attack against the matcher. Let us assume that an adversary uses a Trojan horse for manipulating the matcher and hence to gain access to the system. If an adversary gains access to the matcher, clearly it can give access to whomever it wants and breaks the security of the system. However in this case, even if the adversary can gain access to the proposed system, he cannot obtain the biometric templates since the (2,2)-threshold homomorphic cryptosystem is used. Therefore, the adversary cannot perform decryption since he does not know the private key share of the user (i.e. sk_i^1). Consequently, the proposed system preserves the privacy of the users.

6. Protection against Attack 6 - Attacks against database: An adversary can neither gain access to the proposed system nor obtain biometric template of the users by performing an attack against the database. Let us assume that an adversary aims to gain access to the proposed system by performing this attack. In this case, the adversary must substitute or add data into the database since erasing data from the database does not allow access to the proposed system for the adversary. Let us again assume that the adversary adds data into the database (e.g.

TABLE 7.2: The experimental results

Length of biohash vector (bit)	Modulus(bit)	Enrollment time (s)	Authentication protocol run (s)
0-256	256	0.3488	0.0080
257-512	512	2.1724	0.0341
513-1024	1024	16.3735	0.2180
1025-2048	2048	280.195	1.5528

encrypted version of the biohash of the adversary) or substitute data stored in the database. For such cases, only adding data into the database or substituting data stored in the database is not sufficient to gain access to the proposed system as a legitimate user since the adversary must also successfully execute the proposed authentication protocol. Recall that the verifier verifies the signature on $\mathsf{Sign}_{sk_{U_i}}(\mathsf{Enc}_{pk_i}(r_i), \boldsymbol{Y}, \text{nonce}_{U_i}, \text{nonce}_V)$ at the beginning of the fourth step of the proposed authentication protocol since adversary cannot generate a valid signature without having the private key of the user sk_{U_i}.

On the other hand, let us assume that the adversary aims to obtain the biometric templates of the users by performing an attack on the database. In this case, he can only obtain encrypted versions of the biohashes ($C_i = \mathsf{Enc}_{pk_i}(\boldsymbol{B}_{enroll_i})$) belonging to the legitimate users. However, biohashes are encrypted by using the $(2, 2)$ threshold homomorphic cryptosystem which makes decryption impossible without having both the user's private key share (sk_i^1) and the verifier's private key share (sk_i^2). Consequently, the adversary cannot threaten the privacy of the user even if he steals encrypted data from the database due to the (2,2)-threshold homomorphic cryptosystem.

7. Protection against Attack 7 - Tampering the communication channel between the database and the matcher: An adversary can neither gain access to the proposed system nor obtain biometric template of a user by tampering the communication channel between the database and the matcher. Let us assume that an adversary aims to gain access to the proposed system by performing this attack. The database sends $C_i = \mathsf{Enc}_{pk_i}(\boldsymbol{B}_{enroll_i})$ to the matcher. Thus, when the adversary performs this attack, he can only obtain $C_i = \mathsf{Enc}_{pk_i}(\boldsymbol{B}_{enroll_i})$. On the other hand, the adversary cannot use these data in order to gain access to the proposed system as a legitimate user since

1. he cannot decrypt it because he does not have the user's private key share (sk_i^1) and the verifier's private key share (sk_i^2),

2. he does not have the private key of the user sk_{U_i} for signing the data.

Therefore, the adversary cannot successfully execute the proposed authentication protocol in order to gain access to the proposed system even if he tampers the communication channel between the database and the matcher. Furthermore, he cannot threaten the privacy of the users since he cannot perform decryption on the obtained data $C_i = \mathsf{Enc}_{pk_i}(\boldsymbol{B}_{enroll_i})$ due to the (2,2)-threshold homomorphic cryptosystem. Consequently, the proposed system also preserves the privacy of the users.

8. Protection against Attack 8 - Override response: An adversary can neither gain access to the proposed system nor obtain biometric template of a user by performing an override response attack. The proposed system can easily be updated in order to overcome this attack as described in the step 5 of the proposed authentication protocol. In the proposed authentication protocol, the verifier signs its decision including the nonces generated during the authentication session (i.e. either Sign(Accept, nonce_{U_i}, nonce_V)or Sign (Reject, nonce_{U_i}, nonce_V) and then sends it to the user. Thus, it is explicitly known that the decision is taken by the verifier and it cannot be changed. The signature guarantees authenticity, integrity and origin of the data. The user must provide the signed decision including the nonces generated during the authentication session to the verifier in order to gain access to the system. In this case, the adversary cannot perform the override response attack since he cannot generate the signature of the verifier. Even if he obtains Sign(Accept), he cannot make replay attack since the nonces are also used in the response (e.g. Sign(Accept, nonce_{U_i}). On the other hand, the adversary cannot obtain the biometric template of a user by performing this attack since biometric data is never released at this step.

9. Protection against Attack 9 - Hill-climbing attack: An adversary can neither gain access to the proposed system nor obtain the biometric template of a user by a hill-climbing attack. Let us assume that an adversary aims to perform a hill-climbing attack by using the output of the matcher. In the hill-climbing attack, the adversary improves quality of the synthetically created biometric template by iteratively updating it using a specific modification scheme according to the score given by the matcher. For a one factor authentication system which is only based on biometric data, when the score exceeds a fixed decision threshold, the adversary can gain access to the system. However, the proposed system is a two-factor authentication system where biometric data and private key are needed for successful authentication. Even if the private

keys of the user are compromised, the matcher does not output a score instead outputs just an accept/reject decision (i.e. either Sign(Accept, nonce_{U_i}, nonce_V)or Sign (Reject, nonce_{U_i}, nonce_V) in the proposed system. Consequently, the adversary cannot obtain the scores which are needed for performing the hill-climbing attack. Moreover, the adversary cannot estimate the biometric data of the user since he cannot reach the matching scores and hence the user privacy is not threatened.

7.6 Complexity Analysis

In this section, we discuss the complexity of the THRIVE enrollment and authentication protocols. The complexity of the THRIVE enrollment and authentication protocols cover protocol steps for the round complexity, exponentiation for the computational complexity and messages sent to the parties for the communication complexity. Without loss of generality, we will provide complexity of the THRIVE protocols using the (2,2)-threshold homomorphic Paillier cryptosystem as an instance [115].

The round complexity of the enrollment protocol is only 1. For the computational complexity, the enrollment protocol requires only 4 exponentiations for a user but 1 exponentiation for the server. The enrollment protocol has only 2 encryption and 1 signature and 1 signature verification for the communication complexity.

As regards the authentication protocol, two different encryption schemes are used, i.e., a conventional public key operation for (pk_{U_i}, sk_{U_i}) (e.g., for the conventional encryption scheme like RSA) and $(2, 2)$-threshold homomorphic encryption scheme for $(pk_i, (sk_i^1, sk_i^2))$ (e.g.; for the Paillier encryption). In the authentication protocol, there are only 4 rounds. For the computational complexity of the authentication protocol, the user computes 2 exponentiations for C_i', 2 exponentiations for the decryption shares X and Y, 1 exponentiation for the signature verification, 2 exponentiations for C_i'', and 1 exponentiation for the signature during the first and the third steps of the proposed protocol. Consequently, there are 8 exponentiations computed during the authentication protocol for a user. On the other hand, the verifier computes 1 exponentiation for the signature verification, 1 exponentiation for C_i'', and 4 exponentiations for the decryption shares $D_{sk_i^2}(C_i')$ and $D_{sk_i^2}(C_i'')$. Therefore, there are 6 exponentiations computed during the authentication protocol for the verifier. In total, there are 14 exponentiations computed during

the authentication protocol for the user and the verifier. The authentication protocol has only 1 encryption and 2 signature operations for the communication complexity.

7.7 Implementation of the Proposed System

In this section, we introduce implementation results of the THRIVE enrollment and authentication protocols. We implement the THRIVE enrollment and authentication protocols in Java language on a PC which has Intel 2.27 GHz Core2 Duo processor and 4 GB RAM. In the protocol implementation, we use RSA for the signing scheme and Paillier Threshold scheme for the encryption. In the proposed system, there are two protocol stages such as enrollment and biometric authentication. During the enrollment stage, RSA signing certificates are generated for both the user and the server in order to register a user into the server. Then, a new Paillier threshold scheme is constructed between the user and the server for each enrollment and the required prime numbers are generated in each construction. After that the enrollment protocol is executed. We also utilize a Java library which implements the Paillier Threshold Encryption systems [120] in our implementation. The security of the RSA depends on the difficulty of the factorization of its own common modulus ($n = pq$). The Paillier encryption scheme is based on the problem to decide whether is an n-th residue modulo n^2. This problem is believed to be computationally hard in the cryptographic domain, and is linked to the hardness to factorize its own common modulus $n = pq$ where n is the product of two large primes p and q. In order to provide 80-bit security level for RSA signing, we have to use 1024-bit common composite modulus (512-bit for p and 512 bit for q) for RSA signing. Similarly, the common modulus for Paillier threshold system is also 1024-bit.

In our experiments, we simulate 1000 clients and a single server. We create random credentials for each user. Both the user and the verifier are run on the same PC. A successful authentication protocol run requires **0.218** seconds on average. The average time for the enrollment and the authentication protocol run for different security parameters are depicted in Table II. The enrollment requires much higher than the authentication protocol because of the generation of the common modulus. Recall that the enrollment time shown in Table II includes set-up time and enrollment protocol execution time. Besides, it is clear that an implementation in C would be much faster. Note that the maximum length of the biohash vector can be the number of modulus bits. For instance, if modulus is 1024 bit, the length of the biohash can be from 513 bit (minimum) to 1024 bit (maximum).

7.8 Chapter Summary

In this chapter, we propose a novel biometric authentication system. The aim of the proposed system is to increase security against adversary attacks defined in [4] when an adversary aims to gain access to the system as a legitimate user and protect the privacy of the users by encrypting the biometric templates stored in the database. We design new enrollment and authentication protocols in order to increase the security against attacks reported in the literature and preserve the privacy of the users. The proposed system can be used with any biometric feature extraction method which can produce binary templates or whose templates can be binarized. The biohashing is chosen as an example binary biometric template generation system since it offers satisfactory error rates and fast authentication. The comparison is performed in a randomized domain in the authentication stage and the binary templates (e.g biohashes) are never released. In addition to that, only encrypted binary templates are stored in the database. Since we use the $(2, 2)$-threshold cryptosystem, the verifier cannot, in polynomial time, decrypt the data stored in the database by himself. The user and the verifier can only decrypt the encryption of the binary templates collaboratively. The proposed system can be used in applications where the user and the verifier do not necessarily trust each other. Furthermore, we implement the proposed system and a successful authentication protocol run requires 0.218 seconds on the average. Consequently, it can be stated that the proposed system is sufficiently efficient to be used in the real world applications.

Chapter 8

Conclusion

8.1 Conclusions

In this thesis, we address performance, security and privacy aspects of biohashing methods. We propose new biohashing methods in order to increase authentication performance of the existing random projection based biohashing methods. We develop several biohashing methods in order to improve the performance of the existing random projection based biohashing methods even for the key-stolen scenario. First, we try to find a better projection matrix in order to reduce the Hamming distance between the biometric hash vectors which represent the same user but differ due to variations in the biometric data. We develop a new face image hashing method based on optimal linear transform under colored Gaussian noise assumption. The simulation results show that the proposed biometric hashing method has much better performance in comparison to the random projection based biometric hashing methods in the literature [2, 3]. Furthermore, in general, performance of the proposed method increases with the increasing length of the biometric hash vector. Next, we develop a new face image hashing method based on a proposed technique that we call discriminative projection selection to reduce verification errors. This technique selects the rows of an RP matrix, which is a user dependent dimension reduction matrix, by using the Fisher criterion. In addition, we employ Gaussian mixture model at the quantization step to obtain more distinct face image hash vectors for each user. The proposed method has better performance in terms of EER in comparison to the Ngo *et al.*'s methods [2, 3]. As the length of face image hash vector decreases, the proposed method shows better improvement since the proposed dimension reduction matrix better preserves the pair-wise distances between feature vectors in the reduced dimension subspace in comparison with the traditional random projection matrix.

The best results are usually obtained with 128 or 256 bits. Finally, we propose another new biometric hashing method which depends on Error-Correcting Output Codes (ECOC). We improve the performance of the random projection based biometric hashing scheme by introducing a new quantization method that attempts to optimize biometric hash vectors by using the ideas from ECOC classifiers. The proposed scheme shows superior performance in comparison with Ngo *et al.*'s method. The proposed scheme approximately reduces the EER by half in most of the cases. Furthermore, even in some cases, the proposed scheme perfectly separates the genuine and imposter users with no errors. Discriminative biohashing method in Chapter 4 and ECOC guided biohashing in Chapter 5 have better performance especially in key-stolen scenario in comparison with the random projection based biohashing methods. Discriminative biohashing method performs better for biohash vectors with 64 and 128 bits since this method can select more discriminative bits. As the length of the biohash vector increases, there is not enough discriminative bits to be selected. Thus, the performance of the discriminative biohashing method decreases. On the other hand, the performance of the ECOC guided biohashing increases with the increasing biohash vector length since it is based on optimization of the Hamming distance between the columns and rows of the codeword matrix which contains biohashes in its rows. Dimension reduction and quantization phases introduces verification errors due to information loss at these phases. From the simulation results, it can be stated that ECOC guided biohashing performs better than discriminative biohashing method. We conclude that information loss at the quantization phase is much more than information loss at the dimension reduction phase. Thus, improving the quantization phase more significantly increases the system performance.

In addition to these works, we analyze the security and privacy preservation aspects of the random projection based biohashing methods by performing various attacks against them. We classify threats against these systems into two groups: 1) Security threats, 2)Privacy threats. For security threats, we demonstrate the performance of the proposed methods under key-stolen scenario where an attacker gets the secret key of the legitimate user but does not have her face image. For privacy threats, we perform attacks against random projection based biohashing schemes in order to test its privacy preservation capability via its irreversibility property. We propose new attack methods based on minimum norm solutions for random projection based biohashing schemes for face images. We can reconstruct the biometric data (not exactly the same but similar) of the user by using the minimum norm solutions when the biohash and the secret key of the user are stolen. Therefore, we show that an attacker can threaten the privacy of the users. We also demonstrate that the attacker can also threaten the security of the system by

using the reconstructed image and the secret key of the user even if the user changes her secret key.

Finally, we propose a new biometric verification and template protection system which we call the THRIVE system. The system includes novel enrollment and authentication protocols based on threshold homomorphic encryption in which the private key is shared between the user and the verifier. The system is designed for the malicious attack model where neither of the parties is assumed to be honest. Security of the system is enhanced using a two factor authentication scheme involving the users private key and the biometric data. In the proposed system, only encrypted binary biometric templates are stored in the database and verification is performed via homomorphically randomized templates, hence, original templates are never revealed even during authentication. Since threshold homomorphic encryption scheme is used, a malicious party cannot perform decryption on encrypted templates of the users in the database using a single key. We analyze the security and privacy preservation capability of the proposed system against various adversary attacks. The analysis results show that the system is robust against malicious attacks and preserves privacy of the users. The proposed system is suitable for applications where the user and the verifier do not necessarily trust each other. The system can be used with any biometric modality and biometric feature extraction scheme whose output templates can be binarized. We implemented the proposed THRIVE system and found that a successful authentication protocol run requires 0.218 seconds on average. Consequently, the system can be efficiently used in real life applications.

8.2 Future Work

In this thesis, we address security and privacy issues as well as verification performance of the biohashing methods. We work on face images but different biometric modalities as well as multi-modal biometrics can be studied as future work. For performance aspects, the feature extraction has utmost importance in a biohashing method. Better verification performance can be achieved with better feature extraction methods (e.g. Zernike moments for face images). Fixed length feature extraction methods can be very useful for developing new biohashing methods. Furthermore, the quantization is one of the most critical phases in the biohashing schemes since it causes verification errors. The verification performance of the biohashing methods can be improved by using better quantization methods.

In addition, there may be different security and privacy attack scenarios against biohashing methods. Security and privacy preservation capability of the biohashing methods can be tested with these attack scenarios. For instance, Hellman *et al.* propose rainbow tables in order to crack cryptographic hashed passwords [121]. These tables contain pre-computed values and are used for reversing cryptographic hash functions. Their aim is to recover the plaintext password by using only a hashed value. This idea can be adopted to attacks against the biohashing methods. In this attack scenario, an attacker gets the secret key and the biohash of a user. The user is not aware that her secret key and biohash are stolen. The attacker also has a face image database containing face images which do not belong to any legitimate user. He computes biohashes by using the face images and the secret key of the user. Then, he computes the Hamming distance between the original biohash and the computed biohashes. He uses the face image whose biohash has the minimum Hamming distance with the original biohash. He uses this face image and the secret key of the user to gain access to the system. In another attack scenario, the attacker again gets the secret key and the biohash of a user. The attacker applies exactly the same method in order to find a face image. Consequently, the attacker uses this face image and the secret key of the user to gain access to the system. However, the user notices that her biohash is stolen and changes her secret key for updating her biohash. These attack scenarios can be performed against the biohashing methods as future work. Moreover, security and privacy attacks can be investigated under insider and outsider attacker models. The security and privacy metrics can be proposed in order to evaluate the security and privacy preservation capability of the biohashing methods. The cryptographic protocols which are developed for the THRIVE system can be analyzed by using formal methods.

Finally, new cryptographic protocols can be proposed for biohashing methods according to the usage scenario and for different attack models. Fully homomorphic encryption methods can be used for securing biometric template as well as ensuring privacy.

8.3 Acknowledgments

This work has been supported by the BEAT project 7th Framework Research Programme of the European Union (EU), grant agreement number: 284989. The authors would like to thank the EU for the financial support and the partners within the consortium for a fruitful collaboration. For more information about the BEAT consortium please visit http://www.beat-eu.org.

Bibliography

[1] Anil K. Jain, Karthik Nandakumar, and Abhishek Nagar, "Biometric template security," *EURASIP J. Adv. Signal Process*, vol. 2008, pp. 113:1–113:17, January 2008.

[2] David Ngo Chek Ling, Andrew Teoh Beng Jin, and Alwyn Goh, "Biometric hash: high-confidence face recognition.," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 16, no. 6, pp. 771–775, 2006.

[3] David Ngo Chek Ling, Andrew Teoh Beng Jin, and Alwyn Goh, "Eigenspace-based face hashing," in *Proceedings of the ICBA*, 2004, pp. 195–199.

[4] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle, "An analysis of minutiae matching strength," in *Proceedings of the 3rd AVBPA*, 2001, pp. 223–228.

[5] Yao-Jen Chang, Wende Zhang, and Tsuhan Chen, "Biometrics-based cryptographic key generation," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2004, vol. 3, pp. 2203–2206 Vol.3.

[6] George I. Davida, Yair Frankel, and Brian J. Matt, "On enabling secure applications through off-line biometric identification," in *Proceedings of the IEEE Symposium on Security and Privacy*, 1998, pp. 148–157.

[7] Anil K. Jain, Arun Ross, and Salil Prabhakar, "An introduction to biometric recognition," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, 2004.

[8] 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.

[9] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Systems Journal*, vol. 40, no. 3, pp. 614–634, 2001.

[10] Chris Roberts, "Biometric attack vectors and defences," *Computers & Security*, vol. 26, no. 1, pp. 14–25, 2007.

[11] G. Tomko, "Biometrics as a privacy-enhancing technology: Friend or foe of privacy?," in *Proceedings of the Privacy Laws & Business 9th Privacy Commissioners/Data Protection Authorities Workshop*, 1998.

[12] M. Crompton, "Biometrics and privacy: The end of the world as we know it or the white knight of privacy?," in *Proceedings of the 1st Biometrics Institute Conference*, 2003.

[13] George I. Davida, Yair Frankel, and Brian J. Matt, "On enabling secure applications through off-line biometric identification," in *Proceedings of the IEEE Symposium on Security and Privacy*, 1998, pp. 148–157.

[14] George I. Davida, Yair Frankel, Brian J. Matt, and Ren Peralta, "On the relation of error correction and cryptography to an off line biometric based identification scheme," in *Proceedings of the Workshop on Coding and Cryptography, France*, 1999, pp. 129 – 138.

[15] Andrew B. J. Teoh, Alwyn Goh, and David C. L. Ngo, "Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1892–1901.

[16] Alessandra Lumini and Loris Nanni, "An improved biohashing for human authentication," *Pattern Recognition*, vol. 40, pp. 1057–1065, 2006.

[17] Adams Kong, King-Hong Cheung, David Zhang, Mohamed Kamel, and Jane You, "An analysis of biohashing and its variants," *Pattern Recogn.*, vol. 39, pp. 1359–1368, July 2006.

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

[19] Zhengyao Bai and D. Hatzinakos, "Lbp-based biometric hashing scheme for human authentication," in *Proceedings of the 11th International Conference on Control Automation Robotics Vision*, dec. 2010, pp. 1842 –1847.

[20] Yip Wai Kuan, Andrew B. J. Teoh, and David C. L. Ngo, "Secure hashing of dynamic hand signatures using wavelet-fourier compression with biophasor mixing and 2n discretization," *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, pp. 32–32, Jan. 2007.

[21] Christian Rathgeb and Andreas Uhl, "Iris-biometric hash generation for biometric database indexing," in *Proceedings of the 20th International Conference on Pattern Recognition*, 2010, pp. 2848–2851.

[22] Karl Kümmel and Claus Vielhauer, "Reverse-engineer methods on a biometric hash algorithm for dynamic handwriting," in *Proceedings of the 12th ACM workshop on Multimedia and security*, 2010, MMSec '10, pp. 67–72.

[23] 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 CISST*, 2005, pp. 40–45.

[24] Karl Kümmel, Claus Vielhauer, Tobias Scheidat, Dirk Franke, and Jana Dittmann, "Handwriting biometric hash attack: a genetic algorithm with user interaction for raw data reconstruction," in *Proceedings of the 11th IFIP TC 6/TC 11 international conference on Communications and Multimedia Security*, 2010, pp. 178–190.

[25] Cagatay Karabat, Hakan Erdogan, and Mehmet Kivanc Mihcak, "A face image hashing method based on optimal linear transform under colored gaussian noise assumption," in *Proceedings of the 17th International Conference on Digital Signal Processing*, july 2011, pp. 1 –6.

[26] Cagatay Karabat and Hakan Erdogan, "Discriminative projection selection based face image hashing," *IEICE Transactions*, vol. 95-D, no. 5, pp. 1547–1551, 2012.

[27] Geoffrey J. McLachlan, *Discriminant analysis and statistical pattern recognition*, Wiley series in probability and mathematical statistics. J. Wiley and sons, 1992.

[28] Cagatay Karabat and Hakan Erdogan, "Error-correcting output codes guided quantization for biometric hashing," *IEICE Transactions*, vol. 95-D, no. 6, pp. 1707–1712, 2012.

[29] C.Karabat, M.S. Kiraz, H.Erdogan, S.Kardas, and E.Savas, "Thrive: Threshold homomorphic encryption based secure and privacy preserving biometric verification system," submitted to IEEE Transactions on Systems, Man and Cybernetics, 2013, under review.

[30] A. Adler, "Sample images can be independently restored from face recognition templates," in *Proceedings of the Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2003, vol. 2, pp. 1163–1166.

[31] J. Galbally, C. McCool, J. Fierrez, and S. Marcel, "On the vulnerability of face verification systems to hill-climbing attacks," *Pattern Recognition*, vol. 43, pp. 1027–1038, 2010.

[32] C. Rahtgeb and A. Uhl, "Attacking iris recognition: An efficient hill-climbing technique," in *Proceedings of the ICPR*, 2010.

[33] M. Martinez-Diaz, J. Fierrez, J. Galbally, and J. Ortega-Garcia, "An evaluation of indirect attacks and countermeasures in fingerprint verification systems," *Pattern Recognition Letters*, vol. 32, pp. 1643–1651, 2011.

[34] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1991, pp. 586–591.

[35] Alvin F. Martin, George R. Doddington, Terri Kamm, Mark Ordowski, and Mark A. Przybocki, "The det curve in assessment of detection task performance," in *Proceedings of the EUROSPEECH*, 1997.

[36] Makoto Matsumoto and Takuji Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, Jan. 1998.

[37] T. A. M. Kevenaar, G. J. Schrijen, M. van der Veen, A. H. M. Akkermans, and F. Zuo, "Face recognition with renewable and privacy preserving binary templates," in *Proceedings of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, 2005, AUTOID '05, pp. 21–26.

[38] Horst Feistel, "Cryptography and computer privacy," vol. 228, no. 5, pp. 15–23, May 1973.

[39] Feng Hao, Ross Anderson, and John Daugman, "Combining crypto with biometrics effectively," *IEEE Trans. Comput.*, vol. 55, no. 9, pp. 1081–1088, Sept. 2006.

[40] Sanjay Kanade, Dijana Petrovska-Delacrtaz, and Bernadette Dorizzi, "Cancelable iris biometrics and using error correcting codes to reduce variability in biometric data," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 20-25 June 2009, Miami, Florida, USA*. 2009, pp. 120–127, IEEE.

[41] Ari Juels and Martin Wattenberg, "A fuzzy commitment scheme," in *Proceedings of the ACM CCS'99*. 1999, pp. 28–36, ACM Press.

[42] Sergey Tulyakov, Faisal Farooq, and Venu Govindaraju, "Symmetric hash functions for fingerprint minutiae," in *Proceedings of the ICAPR*, 2005, pp. 30–38.

[43] Yagiz Sutcu, Qiming Li, and Nasir Memon, "How to protect biometric templates," in *Proceedings of the SPIE Conf on Security, Steganography and Watermarking of Multimedia Contents IX*, 2007.

[44] M. van der Veen A. Stoianov A., T. Kevenaar, "Security issues of biometric encryption," in *Proceedings of the Science and Technology for Humanity, 2009 IEEE Toronto International Conference*. 2009, pp. 34–39, IEEE.

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

[46] Cagatay Karabat and Hakan Erdogan, "Trustworthy biometric hashing method," in *Proceedings of the IEEE 17th Signal Processing and Communications Applications Conference*, 2009, pp. 65–68.

[47] Claus Vielhauer, Ralf Steinmetz, and Astrid Mayerhöfer, "Biometric hash based on statistical features of online signatures," in *Proceedings of the 16th International Conference on Pattern Recognition*. 2002, ICPR '02, IEEE Computer Society.

[48] Tee Connie, Andrew Teoh, Michael Goh, and David Ngo, "Palmhashing: a novel approach for cancelable biometrics," *Inf. Process. Lett.*, vol. 93, no. 1, pp. 1–5, Jan. 2005.

[49] R. Fuksis, A. Kadikis, and M. Greitans, "Biohashing and fusion of palmprint and palm vein biometric data," in *Proceedings of the International Conference on Hand-Based Biometrics*, 2011, pp. 1–6.

[50] Stelvio Cimato, Marco Gamassi, Vincenzo Piuri, Roberto Sassi, and Fabio Scotti, "A biometric verification system addressing privacy concerns," in *Proceedings of the CIS*, 2007, pp. 594–598.

[51] Tsutomu Matsumoto, Hiroyuki Matsumoto, Koji Yamada, and Satoshi Hoshino, "Impact of artificial "gummy" fingers on fingerprint systems," *Datenschutz und Datensicherheit*, vol. 26, no. 8, 2002.

[52] Ton van der Putte and Jeroen Keuning, "Biometrical fingerprint recognition: Don't get your fingers burned," in *Proceedings of the CARDIS*, 2000, pp. 289–306.

[53] Julien Bringer, Hervé Chabanne, Gérard D. Cohen, Bruno Kindarji, and Gilles Zémor, "Optimal iris fuzzy sketches," *CoRR*, vol. abs/0705.3740, 2007.

[54] Mauro Barni, Tiziano Bianchi, Dario Catalano, Mario Di Raimondo, Ruggero Donida Labati, Pierluigi Failla, Dario Fiore, Riccardo Lazzeretti, Vincenzo Piuri, Fabio Scotti, and Alessandro Piva, "Privacy-preserving fingercode authentication," in *Proceedings of the 12th ACM workshop on Multimedia and security*, 2010, pp. 231–240.

[55] K. Nandakumar, A.K. Jain, and S. Pankanti, "Fingerprint-based fuzzy vault: Implementation and performance," *Information Forensics and Security, IEEE Transactions on*, vol. 2, no. 4, pp. 744–757, 2007.

[56] Yi C. Feng, Pong C. Yuen, and Anil K. Jain, "A hybrid approach for generating secure and discriminating face template," *Trans. Info. For. Sec.*, vol. 5, no. 1, pp. 103–117, march 2010.

[57] Francis Minhthang Bui, Karl Martin, Haiping Lu, Konstantinos N. Plataniotis, and Dimitrios Hatzinakos, "Fuzzy key binding strategies based on quantization index modulation (qim) for biometric encryption (be) applications," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 118–132, 2010.

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

[59] Walter J. Scheirer and Terrance E. Boult, "Cracking fuzzy vaults and biometric encryption," in *Proceedings of the Biometrics Symposium*, 2007, pp. 1–6.

[60] Andy Adler, "Vulnerabilities in biometric encryption systems," in *Proceedings of the International Conference on Audio and Video based Biometric Person Authentication*, 2005, pp. 1100–1109.

[61] Terrance E. Boult, Walter J. Scheirer, and Robert Woodworth, "Revocable fingerprint biotokens: Accuracy and security analysis," in *Proceedings of the CVPR*, 2007.

[62] Koen Simoens, Pim Tuyls, and Bart Preneel, "Privacy weaknesses in biometric sketches," in *Proceedings of the 30th IEEE Symposium on Security and Privacy*, 2009, IEEE SP'09, pp. 188–203.

[63] 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.

[64] Xuebing Zhou, Arjan Kuijper, Raymond Veldhuis, and Christoph Busch, "Quantifying privacy and security of biometric fuzzy commitment," in *Proceedings of the 2011 International Joint Conference on Biometrics*, 2011, IJCB '11, pp. 1–8.

[65] Koen Simoens, Julien Bringer, Hervé Chabanne, and Stefaan Seys, "A framework for analyzing template security and privacy in biometric authentication systems," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 833–841, 2012.

[66] Tanya Ignatenko and Frans M. J. Willems, "Biometric systems: privacy and secrecy aspects," *Trans. Info. For. Sec.*, vol. 4, no. 4, pp. 956–973, Dec. 2009.

[67] Salil Prabhakar Umut Uludag, Sharath Pankanti and Anil K. Jain, "Biometric cryptosystems: Issues and challenges," in *Proceedings of the IEEE*, 2004, vol. 92, pp. 948–960.

[68] C. Rathgeb and A. Uhl, "Statistical attack against iris-biometric fuzzy commitment schemes," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2011, pp. 23–30.

[69] Tanya Ignatenko and Frans Willems, "Secret rate - privacy leakage in biometric systems," in *Proceedings of the IEEE international conference on Symposium on Information Theory - Volume 4*, 2009, ISIT'09, pp. 2251–2255.

[70] Alex Stoianov, "Security of error correcting code for biometric encryption," in *Proceedings of the Eighth Annual Conference on Privacy, Security and Trust, PST 2010, August 17-19, 2010, Ottawa, Ontario, Canada*. 2010, pp. 231–235, IEEE.

[71] Ee-Chien Chang, Ren Shen, and Francis Weijian Teo, "Finding the original point set hidden among chaff," in *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, 2006, pp. 182–188.

[72] Alisher Kholmatov and Berrin Yanikoglu, "Realization of correlation attack against the fuzzy vault scheme," in *Proceedings of the Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, 2008.

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

[74] Yevgeniy Dodis, Bhavana Kanukurthi, Jonathan Katz, Leonid Reyzin, and Adam Smith, "Robust fuzzy extractors and authenticated key agreement from close secrets," *IEEE Transactions on Information Theory*, vol. 58, no. 9, pp. 6207–6222, 2012.

[75] Yagiz Sutcu, Li Qiming, and Nasir Memon, "Design and analysis of fuzzy extractors for faces," in *Proceedings of the SPIE Optics and Photonics in Global Homeland Security V and Biometric Technology for Human Identification VI*, 2009, vol. 7305.

[76] Thian Song Ong and Andrew Beng Jin Teoh, "Fuzzy key extraction from fingerprint biometrics based on dynamic quantization mechanism," in *Proceedings of the Third International Symposium on Information Assurance and Security*, 2007, pp. 71–76.

[77] Arathi Arakala, Jason Jeffers, and K. J. Horadam, "Fuzzy extractors for minutiae-based fingerprint authentication," in *Proceedings of the ICB'07*, 2007, pp. 760–769.

[78] Xavier Boyen, "Reusable cryptographic fuzzy extractors," in *Proceedings of the 11th ACM conference on Computer and communications security*, 2004, CCS '04, pp. 82–91.

[79] Muchuan Guo Qiming Li and Ee-Chien Chang, "Fuzzy extractors for asymmetric biometric representation," in *Proceedings of the IEEE Workshop on Biometrics (In association with CVPR)*, 2008.

[80] Yagiz Sutcu, Husrev Taha Sencar, and Nasir Memon, "A secure biometric authentication scheme based on robust hashing," in *Proceedings of the 7th workshop on Multimedia and security*, 2005, pp. 111–116.

[81] Andrew Teoh Beng Jin, Kar-Ann Toh, and Wai Kuan Yip, "2^{ns} discretisation of biophasor in cancellable biometrics," in *Proceedings of the ICB*, 2007, pp. 435–444.

[82] Bian Yang, Christoph Busch, Patrick Bours, and Davrondzhon Gafurov, "Robust minutiae hash for fingerprint template protection," in *Proceedings of the Media Forensics and Security*, 2010.

[83] Pim Tuyls, Anton H. M. Akkermans, Tom A. M. Kevenaar, Geert Jan Schrijen, Asker M. Bazen, and Raymond N. J. Veldhuis, "Practical biometric authentication with template protection," in *Proceedings of the AVBPA*, 2005, pp. 436–446.

[84] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft, "Privacy-preserving face recognition," in *Proceedings of the 9th International Symposium on Privacy Enhancing Technologies*, 2009, PETS '09, pp. 235–253.

[85] Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg, "Efficient privacy-preserving face recognition," in *Proceedings of the ICISC*, 2009, pp. 229–244.

[86] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R.D. Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, A. Piva, and F. Scotti, "A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingercode templates," in *Proceedings of the Fourth IEEE International Conference on Biometrics: Theory Applications and Systems*, Sept. 2010.

[87] Florian Kerschbaum, Mikhail J. Atallah, David M'Raïhi, and John R. Rice, "Private fingerprint verification without local storage," in *Proceedings of the ICBA*, 2004, pp. 387–394.

[88] Mehmet Kivanç Mihçak, Yucel Altug, and N. Polat Ayerden, "On minimax optimal linear transforms for detection with side information in gaussian setup," *IEEE Communications Letters*, vol. 12, no. 3, 2008.

[89] Stephane G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, 1989.

[90] P.J. Huber, *Robust Statistics*, Wiley, 1981.

[91] "Cambridge universiy at&t face database," .

[92] Stéphane Pigeon and Luc Vandendorpe, "The m2vts multimodal face database (release 1.00)," in *Proceedings of the AVBPA*, 1997, pp. 403–409.

[93] Berkay Topcu and Hakan Erdogan, "Decision fusion for patch-based face recognition," in *Proceedings of the 20th International Conference on Pattern Recognition*, 2010, ICPR '10, pp. 1348–1351.

[94] Richard O. Duda and Peter E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, 1973.

[95] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 29, pp. 147–160, 1950.

[96] Aleix Martínez and Robert Benavente, "The ar face database," Tech. Rep. 24, Computer Vision Center, Jun 1998.

[97] Daniel B. Graham and Nigerl M. Allinson, *Face Recognition: From Theory to Applications*, Springer, 1998.

[98] "Carnegie mellon university face database," .

[99] Jiawei Han, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers Inc., 2005.

[100] David W. Aha and Richard L. Bankert, "Cloud classification using error-correcting output codes," *Artificial Intelligence Applications: Natural Resources, Agriculture, and Environmental Science*, vol. 11, pp. 13–28, 1996.

[101] Thomas G. Dietterich and Ghulum Bakiri, "Solving multiclass learning problems via error-correcting output codes," *J. Artif. Int. Res.*, vol. 2, no. 1, pp. 263–286, Jan. 1995.

[102] Ludmila I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience, 2004.

[103] J. Kiefer, "Sequential minimax search for a maximum," in *Proceedings of the Amer. Math. Soc.*, 1953, vol. 4, pp. 502–506.

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

[105] Valérie Viet Triem Tong, Hervé Sibert, Jérémy Lecoeur, and Marc Girault, "Biometric fuzzy extractors made practical: a proposal based on fingercodes," in *Proceedings of the 2007 international conference on Advances in Biometrics*, Berlin, Heidelberg, 2007, ICB'07, pp. 604–613, Springer-Verlag.

[106] Arathi Arakala, Jason Jeffers, and K. J. Horadam, "Fuzzy extractors for minutiae-based fingerprint authentication," in *Proceedings of the ICB*, 2007, pp. 760–769.

[107] Scott Shaobing Chen, David L. Donoho, Michael, and A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.

[108] Emmanuel J. Cands, Justin Romberg, and Terence Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, 2006.

[109] Bruce Scheneir, *Applied Cryptology*, John Wiley & Sons, 1996.

[110] S. Schuckers, "Spoofing and anti-spoofing measures," *Information Security Technical Report*, vol. 7, pp. 56–62, 2002.

[111] A. Eriksson and P. Wretling, "How flexible is the human voice?," in *Proceedings of the European Conf. on Speech Technologies*, 1997, pp. 1043–1046.

[112] J. Hennebert, R. Loeffel, A. Humm, and R. Ingold, "A new forgery scenario based on regaining dynamics of signature," in *Proceedings of the IAPR Int. Conf. on Biometrics*. 2007, pp. 366–375, Springer LNCS-4642.

[113] M. Lane and L. Lordan, "Practical techniques for defeating biometric devices," 2005.

[114] Taher El Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Proceedings of the CRYPTO 84 on Advances in cryptology*, 1985, pp. 10–18.

[115] Pascal Paillier and David Pointcheval, "Efficient public-key cryptosystems provably secure against active adversaries," in *Proceedings of the Asiacrypt'99, Lecture Notes in Computer Science*. 1999, pp. 165–179, Springer Verlag.

[116] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen, "Multiparty computation from threshold homomorphic encryption," in *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology*, 2001, EUROCRYPT '01, pp. 280–299.

[117] Oded Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*, Cambridge University Press, 2004.

[118] Aristides Gionis, Piotr Indyk, and Rajeev Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the Proceedings of 25th International Conference on Very Large Data Bases, Edinburgh, Scotland, UK*, September, 1999, pp. 518–529.

[119] A. K. Jain, A. Ross, and S. Pankanti, "Biometrics: a tool for information security," *IEEE Trans. on Information Forensics and Security*, vol. 1, no. 2, pp. 125–143, 2006.

[120] Murat Kantarcioglu and James Garrity, "Paillier threshold encryption toolbox," http://www.utdallas.edu/m̃xk093120/paillier/, 2013.

[121] Martin E. Hellman, "A cryptanalytic time-memory trade-off," *IEEE Transactions on Information Theory*, vol. 26, no. 4, pp. 401–406, 1980.