Biometric Template Security

By

Abhishek Nagar

A Dissertation

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Computer Science

2012

Abstract

Biometric Template Security

By

Abhishek Nagar

With the proliferation of biometric recognition systems, an attacker's benefit in staging a system compromise is also increasing and thus is the need to ensure system security and integrity. This dissertation provides a thorough analysis of the vulnerabilities of a biometric recognition system with emphasis on the vulnerabilities related to the information stored in biometric systems in the form of biometric templates. To motivate the development of techniques to protect biometric templates, we show that fingerprint image can be recovered from a well known fingerprint representation, called the Minutiae Cylinder Codes, with high accuracy. The recovered fingerprint image can in turn be used to create spoof fingers and compromise the systems in which the finger is enrolled.

The techniques to safeguard the biometric templates are categorized into two main groups: biometric cryptosystems and template transformation techniques. While biometric cryptosystems allow binding a secure key to the biometric data to obtain a so called secure sketch from which no information regarding the biometric data or the key can be recovered, template transformation techniques non-invertibly transform the biometric template with the user's password. To analyze and improve the biometric cryptosystems, we study its two main examples: fuzzy vault and fuzzy commitment. Fuzzy vault is used to secure templates represented in the form of a set of points whereas fuzzy commitment is used to secure templates represented as binary vectors. An improved security analysis is provided that takes into account the non-uniform distribution of biometric features. A framework to effectively combine

multiple biometric representations is also proposed. We identify two limitations of a typical biometric cryptosystem, namely, i) linkability i.e. possibility to identify two secure biometric templates generated using the same biometric, and ii) utilization of only simple biometric representations, and develop techniques to overcome them in the context of fuzzy vault.

Various template transformation techniques proposed in literature are studied and the amount of security they impart is evaluated using a comprehensive set of metrics. The analysis of difficulty of template inversion i.e. recovery of the original template given a transformed template is an important element of its security analysis. We develop the template inversion techniques and analyze security imparted by two different transformed templates; one based on point set representation and the other based on binary vector representation. The analysis presented indicates that the two techniques, although generally considered secure, are vulnerable to inversion attacks.

Protection of biometric templates is critical for public acceptability in light of the potential compromise of system security and user's privacy. Equally critical is a rigorous analysis of the security imparted by the techniques developed to protect the biometric templates. We believe that the security analysis presented in this dissertation will streamline the development of new techniques and help in finding a robust solution for protecting biometric data.

Acknowledgments

First of all I would like to express my sincerest gratitude towards my advisor Prof. Anil K. Jain. He has been a great advisor and an unrelenting source of motivation and support throughout the course of my PhD. I have learned a lot from him as a researcher and also as a person par excellence and it will be a life long effort for me to inculcate in me his good qualities. I also thank Prof. Jain for the various professional opportunities I was able to have during the course of my PhD. Thanks are also due to Prof. Jain for the fun filled parties at his home at times much needed. I also express my gratitude towards my senior and also a member of my PhD committee Dr. Karthik Nandakumar. It has been wonderful collaborating with him. With his excellent sense of articulation he has been a great source of clarity in the research we conducted through my PhD. His efforts are undeniable in streamlining my research in this little known and yet very important topic of biometric template security. I would also like to thank Prof. Rong Jin from whom I learned a lot about the topic of machine learning during his wonderful seminars where we had extensive discussions on interesting topics. I also thank Prof. George Stockman for numerous discussions and his advices at many occasions and for the fun filled canoe trips and dinners at his house that I would cherish the most. I would also like to thank my PhD committee member Prof. Hayder Radha, Prof. Richard Enbody and Prof. Pang-Ning Tan for their valuable comments and suggestions on my thesis work.

I would also like to thank Dr. Shantanu Rane, who was my mentor during the three months summer internship at the Mitsubishi Electric Research Laboratory, Boston in 2009, for the wonderful time I had there. Thanks are also due to Dr. Kaushik Josiam and Dr. Farooq Khan who mentored me during my summer internship in 2010 at Samsung Standards Research Laboratory, Dallas who are also currently my fellow

senior researchers since I recently joined them as a full time employee to work on some exciting projects. I would also like to thank Dr. Vivek Raghavan, Mr. Jagadish Babu and Mr. Yashwant Kumar who were my supervisors during my summer stay with the Unique Identification Authority of India in 2011. It was wonderful time I spent there getting to know the intricacies of the world's largest biometric identification undertaking. I would also like to thank Dr. Julien Bringer, Dr. Vincent Despiegel and Dr. Melanie Favre for extensive discussions during my one week visit to Morpho, France.

I would also like to thank the fellow PRIP students and friends with whom I shared a great time: Pavan, Jung-Eun, Soweon, Serhat, Kien, Brendan, Alessandra. Also thanks to post-docs and visitors in the lab with whom I collaborated with on various project: Dr. Jianjiang Feng, Dr. Shengcai Liao, Dr. Heeseung Choi, Dr. Qijun Zhao, Dr. Eryun Liu. I would also like to thank my friends at East Lansing for the good times we shared: Manish, Atha, Vaidy, Mayur, Rahul, Alok.

Special thanks are due to the department secretary Linda Moore for seamlessly handling various logistics involved during the graduate curriculum. Thanks are also due to Norma Teague for help with my various travels during the course of PhD.

Last but not least I thank my grandfather, my parents, and my brother for their unconditional love and support without which this journey would not have been possible.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

The science of identifying a person based on his anatomical or behavioral features was introduced in the late nineteenth century by Alphonse Bertillon, a French policeman. Alphonse developed the first set of tools, that are collectively called the Bertillonage system, to identify repeat offenders[1]. Bertillonage involved measurement of certain anatomical traits of a person mainly including head length, head breadth, length of the middle finger, the length of the left foot, and the length of the forearm as shown in Figure 1.1. These measurements were usually taken from the new convicts and were matched with the measurements already take from the previous convicts to check if the same person was convicted before.

Not long after the Bertillonage system came into practice, Galton [51], Herschel [55], and Faulds [39] noticed the usefulness of the ridge patterns present on our fingertips for identifying an individual. Figure 1.2 depicts the fingerprint details used by Galton [51] for matching two fingerprints. This led to the development of fingerprint matching systems that replaced the tedious and less accurate Bertillonage

[1] A repeat offender is one who has been convicted multiple times on different accounts. Convicts who are repeat offenders are usually given a harsher punishment in order to de-motivate them to commit a crime again [99]. As a response, the convicts, on the other hand, try to conceal their identities to evade the harsh punishment.

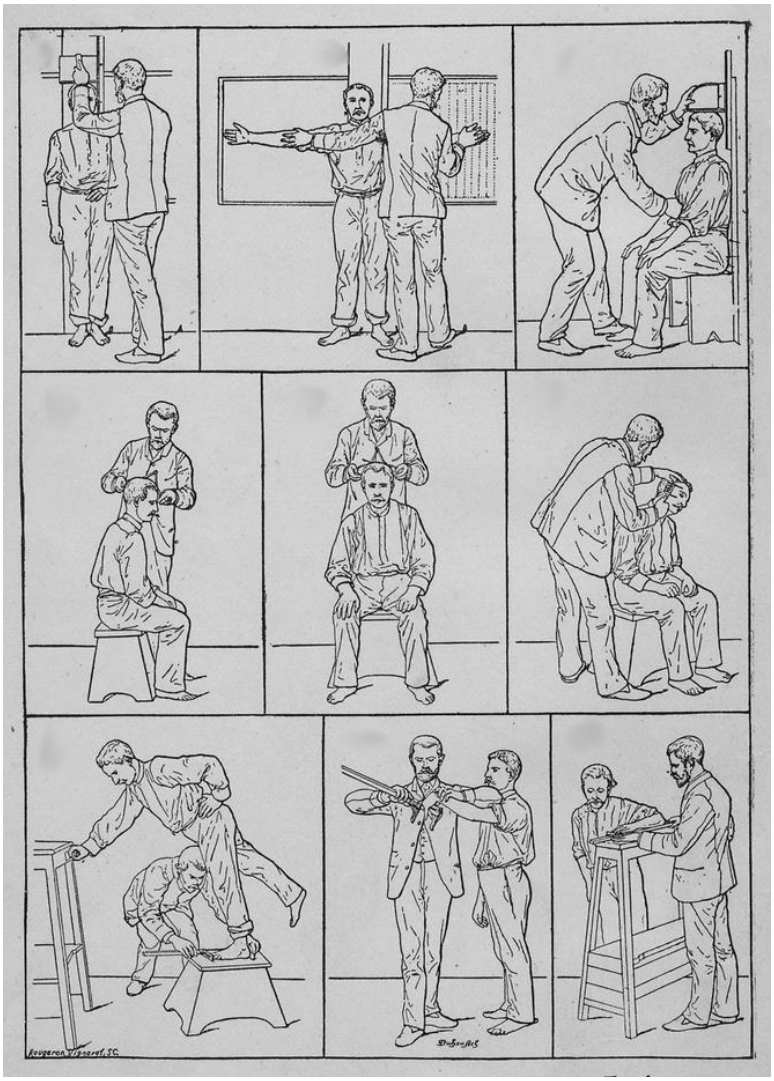


Figure 1.1: Instructional diagram for Bertillonage: the first biometric recognition system [1]. From left to right and then top to bottom the figures show measurement of height, reach, trunk, length of head, width of head, right ear, left foot, left middle finger, and left forearm.

(a)

(b)

Figure 1.2: Example of features extracted from a fingerprint as depicted in [51]. (a) impressions of the fore and middle fingers of the right hand of Sir William Herschel (one of the first British officers in India to use fingerprints on contracts, see [56]), and (b) the corresponding extracted features.

system. Initially, the fingerprints were manually matched by the experts but with the progress in computing technology, systems were developed in 1960's to automate the processing (acquisition, matching and storage) of fingerprints [17,48]. In addition to fingerprints, automatic processing of other personal traits such as palmprints [9], face [16], iris [33], etc. were also developed in parallel. These body traits used for identifying an individual are known as biometric traits and the science of identifying an individual based on his anatomical and behavioral traits is known as biometric recognition or biometrics.

With the development of techniques to process biometric traits in real time, biometrics is also being used as a means of user authentication in applications such as computer log-in or gaining access to a building. Traditionally, user authentication is performed based on passwords (*something you know*) or tokens such as smartcards (*something you have*). These techniques are, however, inconvenient and less secure since passwords can be forgotten or guessed and the tokens can be lost or stolen. Biometrics, on the other hand, provides a convenient means of authentication as it is based on *something you are* that cannot be lost or forgotten. Currently, biometric based recognition systems are being extensively used in a wide range of applications spanning governmental, forensic, and commercial sectors. As an example, the Government of India is implementing a system to capture and store multiple biometric traits (face, fingerprints and iris) from its population of more than 1 billion individuals for the purpose of issuing them a unique identification number (UID) [102]. The worldwide biometrics industry is also expected to grow steadily from an annual revenue of 2 billion USD in 2009 to 11 billion USD in 2017[2]. In most of these applications biometrics is mainly used to either identify an individual from an existing database (called identification) or verify the identity claimed by a user (called verification) based on the acquired biometric trait.

[2] `http://www.acuity-mi.com/FOB_Report.php`

1.1 Biometric System

A biometric recognition system, or simply a biometric system, is a pattern recognition system that recognizes an individual based on his biometric traits. A biometric recognition system consists of four main modules: (i) sensor that captures samples of a biometric trait, (ii) feature extraction module that extracts certain salient features from the biometric sample captured by the sensor, (iii) system database that stores the features extracted by the feature extraction module, and (iv) matcher module that matches the features extracted from the biometric samples with the features stored in the system database. See Figure 1.3 for an illustration of a typical biometric recognition system.

For the sake of brevity, we shall adopt the following terminology related to a biometric system:

- **Biometric trait**: An anatomical or behavioral traits of an individual that is processed and matched for person verification. Examples include fingerprint, face, and iris.

- **Biometric instance**: A specific instance of a biometric trait such as the left eye or the right index finger.

- **Biometric sample**: The snapshot of a specific instance of an individual's biometric captured by a biometric sensor.

- **Biometric template** (or simply template): The features extracted from the biometric sample acquired during user enrolment that are stored in the system. See Figure 1.4 for sample biometric templates extracted from fingerprint, face and iris traits.

- **Biometric query** (or simply query): The features extracted from the biometric sample provided by a user during authentication to be compared with the

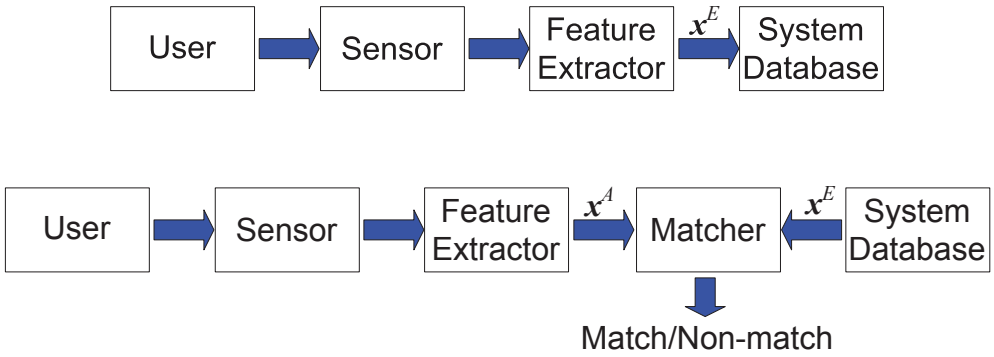


Figure 1.3: Enrolment (top) and authentication (bottom) stages of a typical biometric recognition system. $\mathbf{x}^E$ denotes the feature vector that is stored as a template in the system database during user enrolment. $\mathbf{x}^A$ denotes the query feature vector. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this dissertation.

templates stored in the database.

- **System threshold**: The minimum value of similarity between the query and a template such that the query can be accepted by the system as genuine.

1.1.1 Modes of Operation

A typical biometric system operates in two main modes: enrolment and authentication. In the enrolment mode, the system captures the biometric samples from the user and stores the features extracted from the sample in the system database as a biometric template, $\mathbf{x}^E$, along with the identity of the user, I. Depending on whether the biometric system is being used for identification or verification, the authentication stage is implemented differently. In a verification system, the user provides his identity, I, along with the biometric sample to the system. The features, $\mathbf{x}^A$, extracted from the query biometric sample is matched only with the template, $\mathbf{x}^E$, stored against the claimed identity and the system declares a match if the match score is greater than the system threshold and declares a non-match, otherwise. Most of the commercial biometric recognition systems operate in verification mode where

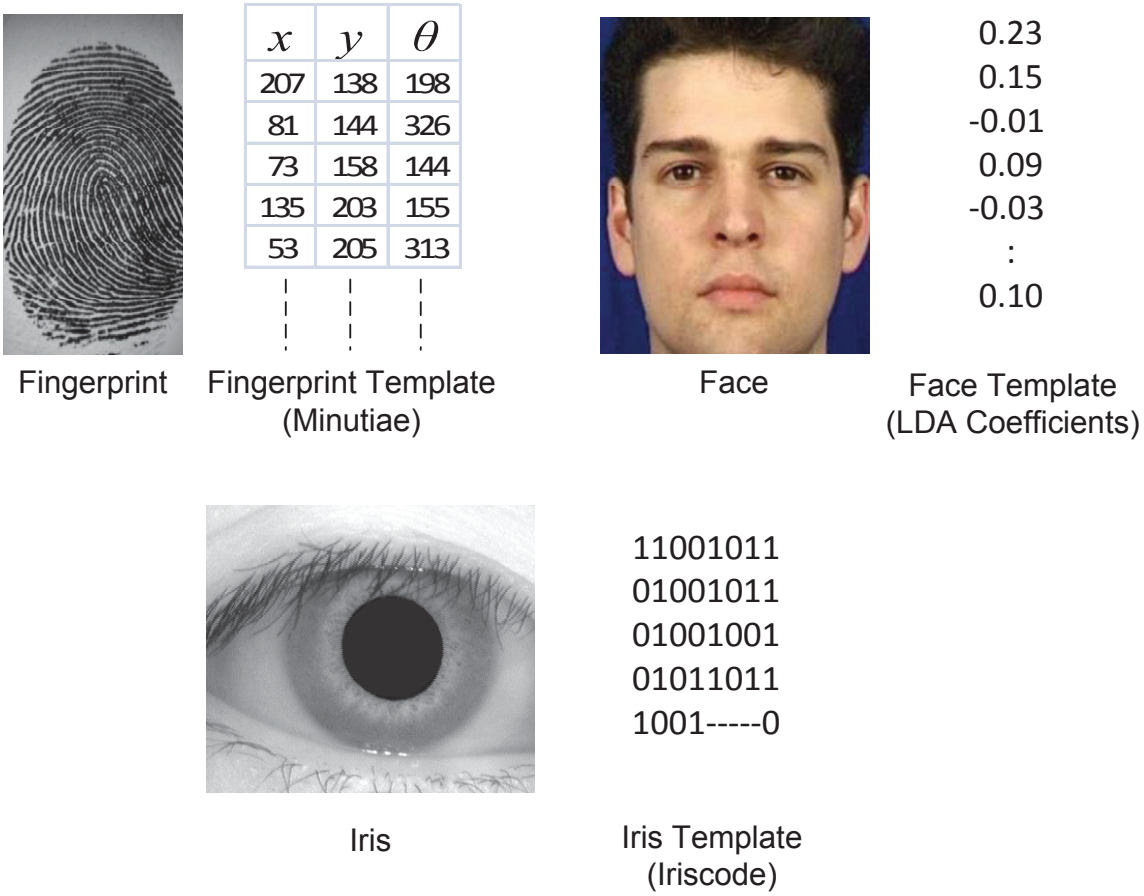


Figure 1.4: Biometric templates extracted from fingerprint, face and iris. Fingerprints are usually represented using set of points marking the endings and bifurcations of ridge lines called minutia which is encoded as a 3-tuple (x, y, θ) with x and y representing the location of the minutia and θ representing the direction of minutia. Face image is usually represented using a vector of Linear Discriminant Analysis (LDA) coefficients. Iris image is usually represented using the binarized responses of Gabor filters, typically called the Iriscode.

the user is asked his identification number before biometric matching.

In an identification system, the user provides only the biometric sample to the system without claiming any identity during authentication. The query thus acquired by the system is matched with all the templates stored in the system database. If one of the templates in the database matches the query, a match is declared; otherwise the system declares a non-match. In addition to a match/non-match decision, the system may also output the identity of the matched user. In certain implementations, the system generates a list of candidate identities from the database ordered according to their similarity to the query. The biometric templates stored against the candidate identities are then manually matched with the query in order to identify the true match. Identification systems are most commonly used in governmental applications where the objective is to identify an individual by matching his biometric traits against an available database. Examples of identification systems include, FBI's IAFIS [101] and the UID system by the Government of India [102]. In fact, the first biometric system, i.e. Bertillonage, also operated in identification mode. However, in this thesis we shall mainly focus on verification mode of operation because the security of biometric templates is of greater concern in the case of relatively less controlled commercial applications where verification is the most common mode of authentication. Furthermore, some of the techniques developed here can be extended for identification scenario while taking advantage of the fact that identification essentially involves verification of the query biometric with each of the identities stored in the database.

1.2 System Vulnerabilities

Biometric recognition systems are prone to deliberate attacks as well as inadvertent security lapses that can lead to illegitimate intrusion, sabotage [3] or theft of sensitive information such as the biometric templates of the users enrolled in the system. The various factors that lead to such security lapses typically belong to one of the following four categories: intrinsic failures, administrative privileges, non-secure infrastructure, and access to biometric data.

1. **Intrinsic failures**

   Due to the non-rigid and genetic nature of the biometric traits and variations in the imaging conditions, the captured biometric images and thus the features extracted usually exhibit large inter-class similarities and intra-class variabilities. As an example, the face images of two identical twins are very similar to each other and this may lead to an incorrect decision while verifying the identity of one of the twins. The rate at which a biometric system incorrectly matches two unrelated biometric templates is called the false match rate (FMR) of the system. The term FMR is also sometimes referred to as the false accept rate (FAR) of the system. While the former value quantifies the frequency of false match decisions, the latter quantifies the frequency with which an impostor is accepted by the system. Note that the two quantities will be different if a user is allowed multiple attempts to provide biometric data during authentication. However, for the course of this dissertation, we assume only single matching attempt per authentication, thus FAR=FMR. A biometric recognition system, on

[3] Note that in a sabotage or a denial of service attack, the aim of the attacker is to render the system unusable for the legitimate users. One benefit to the attacker by staging this kind of attack is that it will force the system administrator to invoke the exception processing routines that are, in general, easier to compromise.

the other hand, may also fail to match two biometric templates extracted from the same biometric due to large intra-class variation. See e.g. Figure 1.5 where two very different fingerprints obtained from the same finger are shown. Such errors are measured using the false non-match rate (FNMR) or false reject rate (FRR) of the system. The FRR measures the number of times a genuine user is rejected by the system irrespective of the number of attempts. However, due to the assumption of single match per authentication attempt, FRR=FNMR. The term genuine accept rate (GAR) is also commonly used to quantify the system performance where GAR= 1−FRR.

These failures of a biometric system to correctly identify an individual can be leveraged by an adversary to gain illegitimate access to the assets protected by a biometric system. An adversary can simply present any available instance of the required biometric trait and expect it to be accepted by the system with a non-zero probability. Such an attack is also referred to as a zero effort attack in literature. This is because the attacker does not make any effort in addition to what is expected from a legitimate user in order to intrude into the system.

Further, if it is very difficult to capture the required biometric trait with an acceptable image quality from a user, the system would practically be unavailable for that user. An example would be an individual wearing a band-aid on his injured finger trying to authenticate himself using a fingerprint recognition system. Exception processing routines are usually executed to authenticate such users based on their identity documents such as a passport or a driver license. An adversary can leverage the insecure nature of exception processing routines to get accepted by the biometric system. To reduce the susceptibility of a biometric system to intrinsic failures, the system threshold should be appropriately tuned. Improvements in feature processing to obtain highly salient

Figure 1.5: Intra class variation among fingerprints. Two fingerprints from the same finger having large variation in the portion of the finger printed.

biometric templates, combining multiple biometric traits (multibiometrics), and combining biometrics with other forms of authentication such as password or smartcards (multifactor authentication) can also reduce the intrinsic failures.

2. **Administrative privileges**

   The system administrators usually have the privileges to make exceptions for the individuals whose biometric traits cannot be acquired by the system possibly due to some injury or disease. This functionality of the system can be abused by an attacker by colluding with or coercing a system administrator to let themselves enrolled or accepted as a legitimate user. An enrolled user can also, either under force from an adversary or inadvertently, keep the access to the system open (e.g. in a biometric door lock system) for the attacker. In order to limit such vulnerabilities, the system administrator should be kept

anonymous and audit trails should be frequently monitored in order to identify any suspicious activity. Other measures such a continuous user authentication [93] can also be used in specific scenarios.

3. **Non-secure infrastructure**

   An attacker can also exploit the hardware infrastructure of a biometric system. Attacks against the biometric infrastructure can be categorized into four main categories:

   (a) **Attacks at user interface**

   The user interface of a biometric system essentially comprises of a sensor that senses the user's biometric trait and provides the sensed data in digital form to the feature extraction module. With the intention of invoking the less secure exception processing routines, an adversary can damage the user interface. The adversary can also try to masquerade as a legitimate user by presenting a replica of the user's biometric trait. In case an adversary is trying to evade an identification system, he can alter his own biometric trait to avoid being matched to his enrolled template.

   In order to avoid the above attacks, first, the sensor should be made robust to any attempt of physical damage. Second, liveness detection techniques should be implemented in order to detect presentation of spoof biometrics. However, design of an effective liveness detection technique that incurs low cost and operates in real time is still a challenge. See Table 1.1 for a list of spoof detection techniques. And third, techniques to detect altered biometrics should be implemented [140].

   (b) **Attacks at interface between modules**

   If the communication channels between various modules of a biometric system are inadequately secured, an adversary can potentially stage a

Technique	Biometric Trait	Property used
Parthasaradhi et al., 2005 [97]	Fingerprint	Perspiration pattern
Antonelli et al., 2006 [12]	Fingerprint	Skin distortion
Setlak, 1999 [114]	Fingerprint	Electrical resistance
Nixon and Rowe, 2005 [94]	Fingerprint	Spectral characteristics
Li et al., 2004 [76]	Face	Fourier spectrum properties
Kollreider et al., 2005 [72]	Face	Motion of different parts
Jee et al., 2006 [66]	Face	Eye movement
Daugman, 1999 [34]	Iris	Photonic, spectrographic (red eye, purkinje) and behavioral properties (hippus, light reflex)
Lee et al., 2006 [74]	Iris	Purkinje reflection

Table 1.1: Different liveness/spoof detection techniques for fingerprint, face and iris.

man-in-the-middle[4] attack to intercept or replace the information being transmitted. The malicious information injected into the system can either allow the adversary to steal the biometric template, gain illegitimate access to the system or bar any legitimate user from accessing the system. One way to avoid such an attack is by cryptographically verifying all the information sent from one module to the other by techniques using e.g. RSA cryptosystem.

(c) **Attacks on software modules**

An attacker can potentially modify or replace a software module using a computer virus injected into the system during certain administrative operation to force the module to output the values desired by the attacker. An attacker can also leverage any algorithmic loopholes in the software. Consider an example where the matching module of a fingerprint recognition system always declares a match if the sensor area is covered by a

[4]In a typical *man-in-the-middle* attack, the attacker intercepts all the communications between two communicating entities and can replace the data being exchanged with any desired data.

sheet of white paper. While this vulnerability might not affect the normal functioning of the system, an adversary can exploit this loophole for gaining illegitimate access to the system without being noticed. To protect the system against such attacks, the software modules should be thoroughly studied and analyzed for all possible inputs and secure code execution practices [113] should be enforced.

(d) **Attacks on the template database**

An attacker can potentially read or even replace the templates stored in the system database with a desired template (e.g., attacker's own template) in order to gain illegitimate access. Note that access of the user's biometric data by an adversary is a compromise of user's privacy [83, 135]. Furthermore, the accessed template can be used to generate spoof biometrics and compromise biometric systems in which the same user is enrolled.

A number of techniques have been proposed to limit such attacks as discussed in Section 1.4 as well as the subsequent chapters.

4. **Access to biometric traits**

The fourth vulnerability of a biometric system arises from the fact that most of the biometric traits such as face, fingerprint, palmprint, voice, and even iris are not secrets. These biometric traits can be covertly captured without the knowledge of the subject using a camera or a microphone. The captured biometric data can be used for various nefarious purposes as mentioned in Section 1.3. Although it is not very difficult for an adversary to access biometric traits of individuals in public, it is usually difficult to ascertain the digital identity of the person whose biometric data has been captured. Furthermore, it is usually easier and safer for an attacker to hack into a system database and obtain biometric information about a large number of individuals along with their identifying information. The security of templates stored in the biometric systems

is thus important.

1.3 Consequences of Template Compromise

There are a number of different ways an adversary can use the information available in the templates stored in a database.

1. **Database linkage**

   The adversary can ascertain if two templates from different databases belong to the same person. This allows the adversary to track the activities of a user. Furthermore, different databases may contain different pieces of information about an individual, a linkage across different databases will thus allow an adversary to consolidate such information enabling him to stage a more severe identity related attack.

2. **System Intrusion**

   There are three main ways in which an adversary can use the stolen biometric templates to gain illegitimate access to a biometric system: template replay, spoof construction and targeted false accepts. An adversary can possibly inject the templates stolen from a biometric system directly into the system in which the same user is enrolled. The biometric image can also be recovered from the templates, e.g. by using a hill climbing [8] attack[5], and can be used to prepare spoof biometrics. A spoof can then be used by an adversary to gain illegitimate access to the biometric systems. Finally, if an adversary can access the biometric data in a system database, he can determine if a user's biometric trait is *similar* to his own. With this information in hand, the adversary can

[5] In a hill climbing attack the attacker essentially implements an iterative optimization algorithm to recover the original template where the fitness function is determined by the matching score between the transformed version of the current estimate of the original biometric and the stored template.

easily masquerade as this user to gain illegitimate access to the system.

3. **Recovery of subject's medical condition**

   It has been shown in the literature that certain medical conditions about the user can be recovered from his biometric template [83,134]. It is postulated that this information can be used to deny insurance or employment to individuals showing signs of certain disorder.

To limit database linkage, a template generated from an individual's biometric should not be matchable to any template previously generated from the same biometric. We refer to this characteristic of a template protection technique as non-linkability or cancelability. In order to limit the recovery of any private medical information of a user from the stored biometric templates, template protection techniques should be developed that make it difficult to recover the original sample from the protected template. We refer to this characteristic of template protection techniques as non-invertibility. Note that non-invertible biometric templates are also resilient to system intrusion as it is not possible to construct a biometric spoof without inverting the template.

A protected template should thus be non-invertible as well as its non-linkable. However, there is often a trade-off between these security characteristics of a protected template and the matching performance of the system particularly characterized by the GAR. In addition to template protection techniques liveness detection techniques are also instrumental in thwarting certain attacks such as template replay and spoof presentation.

1.4 Template Protection Techniques

Typically, passwords are used to protect digital data, however, there are many reasons why passwords are not the best way to protect biometric templates. First, one

of the advantages of biometrics is their convenience and requiring the user to remember passwords that would be used to decrypt their biometric templates during each authentication attempt would undermine the convenience provided to the user as he would have to remember and provide a complex password during every authentication attempt. Second, the strength of security imparted by a password is not sufficient to protect biometric templates.

Thus a number of specific hardware and software solutions have been proposed to protect biometric templates. The hardware solutions mainly involve designing a "closed" recognition system, where the template never leaves a physically secure module and thus cannot be inverted or linked. An example of such a solution is a commercial product called privaris PlusID [2]. In this product, the complete biometric system including the biometric sensor is encased in a keyfob-sized device. During enrolment, the device generates a template from the biometric sample captured from the user and stores it inside the device. And during authentication, if the query captured from the user matches with the stored template, the device transmits a key to, say, an access control system (e.g., a garage door) that can open or close based on the key it receives. A common name for similar devices is "system on card". Another similar system, called "match on card" hosts a template database and the matcher inside a small physically secure module where, during authentication, the biometric captured by an external entity is sent to the system for matching. One of the main limitations of the hardware based solutions is that they are expensive and inconvenient mainly because a user has to carry them and are prone to being lost.

In the software based techniques, the biometric data is usually combined with some external key, such as a password or a system generated random number and the resultant data is stored in the system database instead of the original biometric template. It is expected that the protected template reveals little information about the original template. Based on the way in which the matching is performed, the soft-

ware based template protection techniques can be divided into three main categories: Encryption, Biometric cryptosystems, and Template transformation. See Figures 1.6, 1.7, and 1.8 for the schematic representations of the three categories.

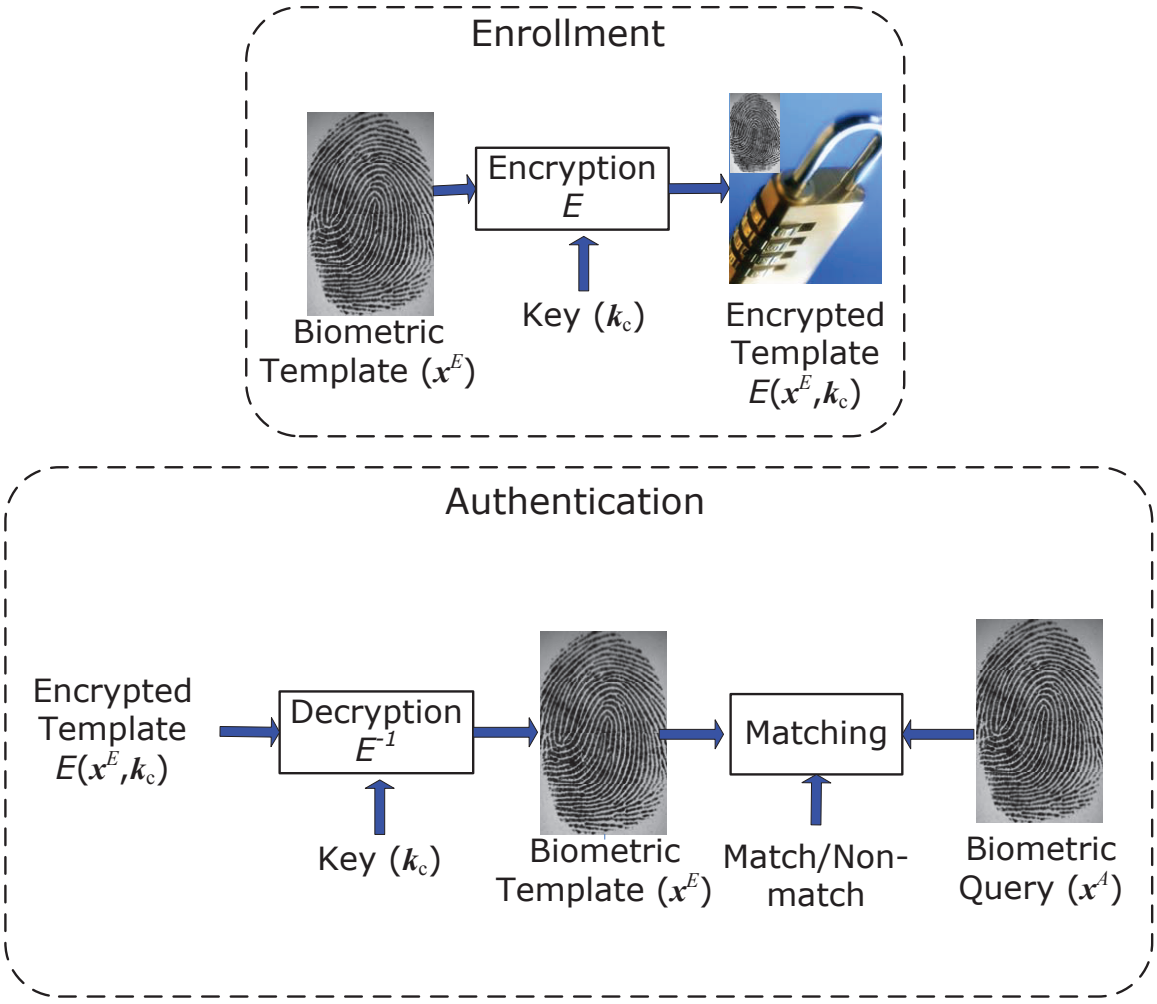


Figure 1.6: Schematic diagrams for enrolment and authentication stages of encryption.

1. **Encryption**

   In encryption based techniques, the biometric template is encrypted using an encryption key, possibly derived from a password, during enrolment. During authentication, the stored data is decrypted using the corresponding decryption

key and is matched with the captured query. See Figure 1.6. Two different kinds of encryption techniques can be used: symmetric and asymmetric. The symmetric encryption, such as the Advanced Encryption Standard (AES) [6], is the simplest form of encryption where the decryption key is the same as the encryption key. In the case of asymmetric encryption, the encryption key is different from the decryption key and it is computationally hard to obtain one from the other. Since the encryption key may be discarded after constructing the secure template, the adversary would not be able to replace the existing encrypted templates even if he steals the decryption key. One of the main limitations of encryption based techniques is insecure key management since the decryption key is exposed to the system during each attempt to authenticate and thus can be easily stolen by the adversary. The advantage, however, is that any sophisticated matching procedure can be employed thereby preserving the matching accuracy.

2. **Biometric cryptosytem**

The second approach to protect a biometric template is using a biometric cryptosystem. In a typical biometric cryptosystem, a key is associated with the biometric data to obtain the so called secure sketch or helper data that does not reveal any information about the biometric template. During authentication, the query is used to recover the original biometric template from the helper data and the exact recovery of the original biometric data is verified to authenticate a user. The main advantage of biometric cryptosystem is that exact recovery of original biometric data allows its use as an encryption key in another cryptosystem e.g. a cryptosystem used to secure online transactions. See Figure 1.7.

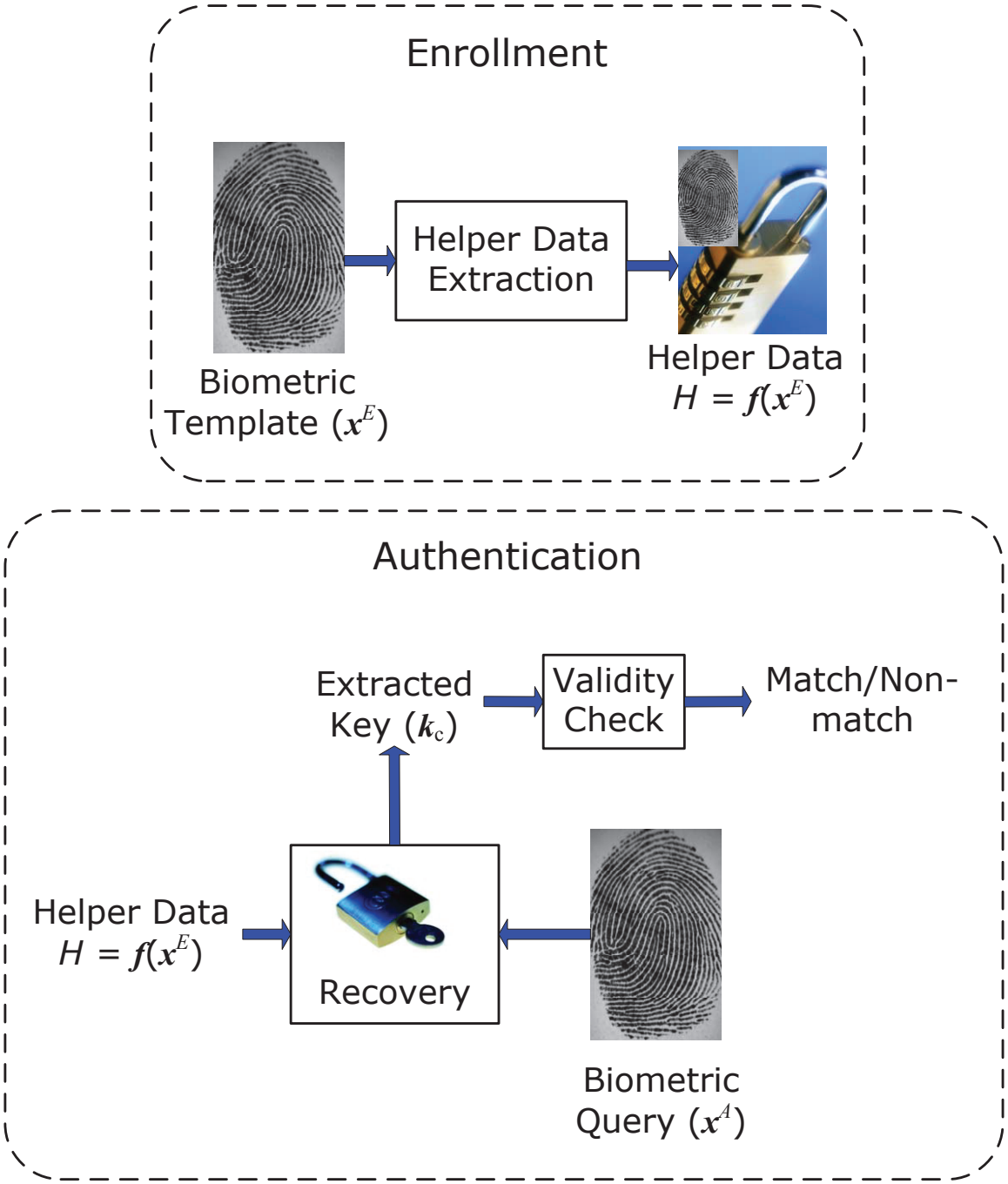


Figure 1.7: Schematic diagrams for enrolment and authentication stages of biometric cryptosystems. Note that in certain constructions the "Helper Data Extraction" module may not involve introduction of a system key during enrolment thus a key is not explicitly shown in the enrolment part of the schema.

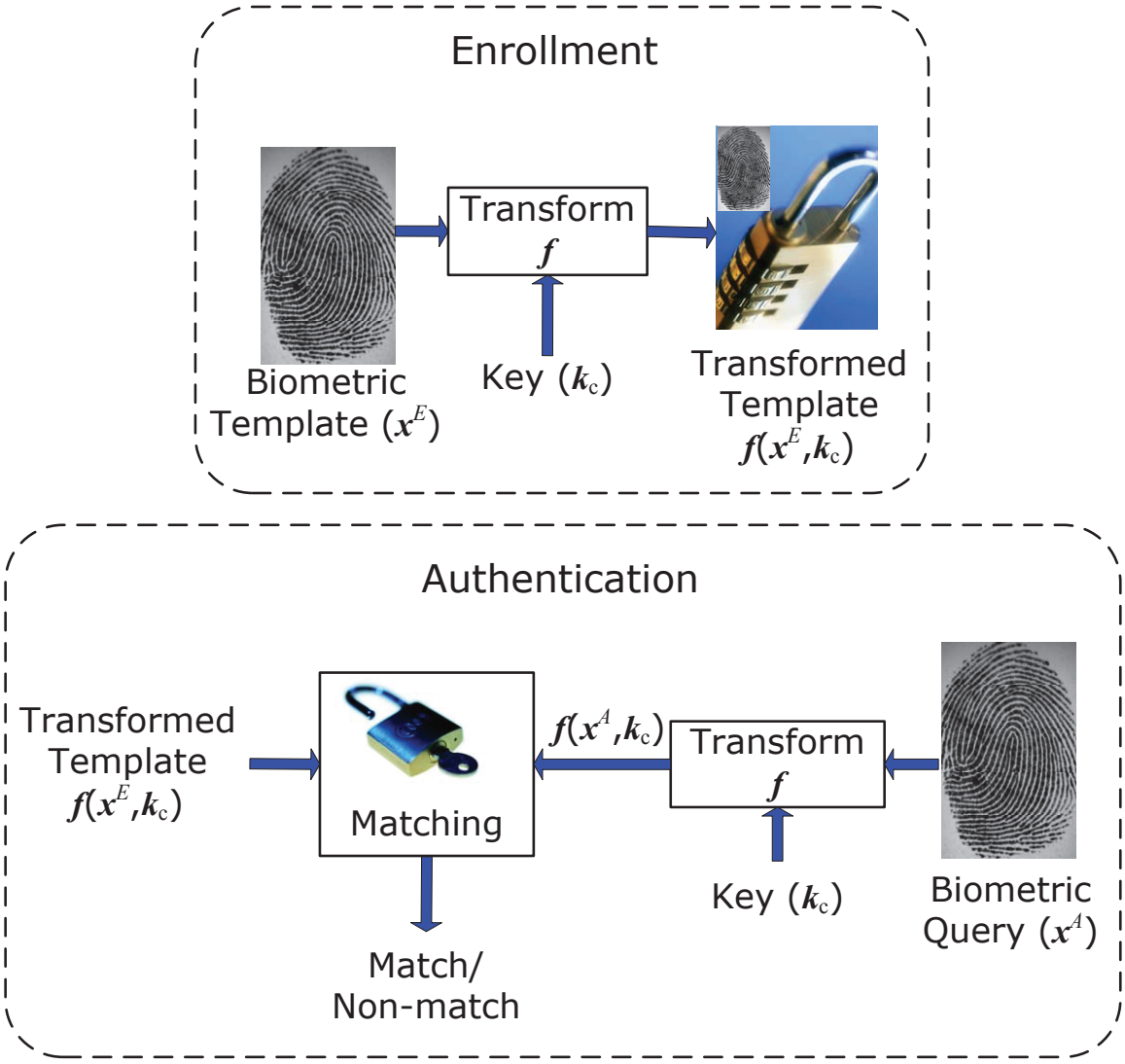


Figure 1.8: Schematic diagrams for enrolment and authentication stages of template transformation.

3. **Template transformation**

In a template transformation technique, during enrolment, the template is transformed using the user's password and during authentication, the query is also transformed using the same password before being matched with the transformed template. See Figure 1.8. Usually, geometric transformations involving projection onto a new space determined by the password are applied to the biometric features. The main advantage of template transformation techniques is that if the user transforms his biometric on a separate personal device and sends only the transformed template to the biometric system, the original biometric is never revealed in the system. The main limitation of such techniques is the loss in performance when the attacker has access to the user's password. This is because the transformation usually leads to a loss of discriminative information available in the biometric data. The overall discriminative information in the protected template may, however, be increased due to the contribution from the user specific password. A class of client-server based cryptographic protocols for biometric authentication that avoid exposure of the biometric data to the server during authentication may also be considered as template transformation techniques. Typically, homomorphic encryption is used in such protocols that allow the template matching to be performed in the encrypted domain at the server. See [19, 118, 130] for examples of such protocols. There are two main differences between these homomorphic cryptographic protocols and the geometric transformation based techniques. First, the homomorphic cryptographic protocols preserve the matching accuracy while geometric transformation may degrade the matching accuracy compared to the case when no technique is used to protect the templates. Second, as in the case of encryption techniques, the cryptographic protocols also require decryption during each authentication

attempt.

Note that the three techniques discussed above are independent in nature and can be used in any combination. For example, the templates protected using either the template transformation or biometric cryptosystem can be further encrypted and a transformed template can be secured using biometric cryptosystem which in turn can also be encrypted. Various distinctive characteristics of these three techniques are also enlisted in Table 1.2. In this dissertation we shall mainly focus on techniques related to geometric transformation of templates and the biometric cryptosystems, primarily because of a number of open research issues in these two techniques related to handling biometric data with sophisticated representation having large intra-class variation. A number of techniques to protect biometric templates have been proposed till date but there has been insufficient effort towards developing a robust analysis of security imparted by these techniques. Note that assurance of security of the stored biometric templates is the primary reason for development of these techniques. The need for thorough analysis of techniques developed to provide information security is also highlighted by Anderson and Moore [10] who note that

Akerlof's 'market for lemons' explains why so many information security products are poor: buyers are unwilling to pay a premium for quality they cannot measure.

A major portion of this thesis is thus devoted towards developing a comprehensive analysis of the security imparted by the available template protection techniques.

1.5 Contributions

The contributions of this dissertation are as follows:

1. Invertibility analysis of a well-known fingerprint minutiae descriptor, namely, the Minutiae Cylinder Code.

	Encryption	**Biometric Cryptosystem**	**Template Transformation**
Description	Encrypt the template; decrypt before authentication	Bind a key to biometric to obtain secure sketch; recover the key or original biometric for verification	Transform template using a password; query also transformed before matching
Match criteria	Score (original biometric)	Key recovery	Score (transformed)
Access to biometric	During authentication attempt	After accept decision	Never
User's Responsibilities	Provide biometric	Provide biometric	Provide biometric and password
System's Responsibilities	Store key	Keep key safe after accept decision	None
Main advantage	Performance preservation	Provides key management	Ensures non-linkability
Main limitation	Key management	Linkability	Weak security
Available implementations	[19, 118, 130]	fingerprint [20, 91, 139], face [20, 44, 71], iris [54, 75]	fingerprint [124], face [46], iris [142]

Table 1.2: Characteristics of software based template protection techniques.

2. New metrics to perform security (invertibility and linkability) analysis of template transformation schemes such as biohashing and cancelable fingerprint templates.

3. Invertibility analysis of well-known biometric cryptosystems such as fuzzy vault and fuzzy commitment and evaluation of the security-GAR tradeoff in such systems.

4. Enhancing the security-GAR tradeoff in biometric cryptosystems through feature level fusion of multiple biometric traits.

5. Improving the non-linkability and non-invertibility of a fingerprint-based fuzzy vault through the incorporation of passwords and texture-based minutiae descriptors, respectively.

1.6 Thesis Organization

In chapter 2 we discuss the possibility of reconstructing fingerprint images from descriptor based templates. We develop inversion techniques for a well know fingerprint template, namely the minutiae cylinder codes (MCC). In chapter 3, we shall discuss the biometric cryptosystems in detail and provide measures to evaluate their effectiveness. The fourth chapter discusses a framework to incorporate multiple modalities in a single biometric cryptosystem. Here we note that the proposed technique performs significantly better than a simple cascade implementation. In chapter 5, we shall focus on fingerprint fuzzy vault to secure minutiae. We identify some of the limitations of the available fuzzy vault constructs and suggest improvements that allow inclusion of user's password and additional features related to the fingerprint ridge information. In the sixth chapter, we provide a comprehensive set of metrics to analyze the security imparted by a template transformation technique. The final chapter summarizes our

contributions and provides suggestions for future work.

Chapter 2

Fingerprint Template Inversion

2.1 Introduction

Biometric data is typically stored in the form of biometric templates that consist of salient and efficiently matchable features extracted from the biometric signal or image captured during user enrolment. If the original biometric image, e.g. a fingerprint or a face image from which the template is derived, can be reconstructed from a stolen template, a physical replica of the biometric, called a spoof biometric, can be constructed thereby compromising the system security. Recovery of biometric image is also a concern due to potential of deriving sensitive personal information from a biometric image. Thus, from the system security perspective, the system designer should ensure that it would be extremely hard to reconstruct the biometric image from its stored template. From the matching accuracy perspective, however, the template should contain as much individualizing information as is available in the associated biometric image. In other words, there should be very little loss of discriminatory information while extracting the template from a biometric image. These two are competing requirements and it is important to find a template configuration that satisfies both these requirements as best as possible. We feel that a thorough analysis

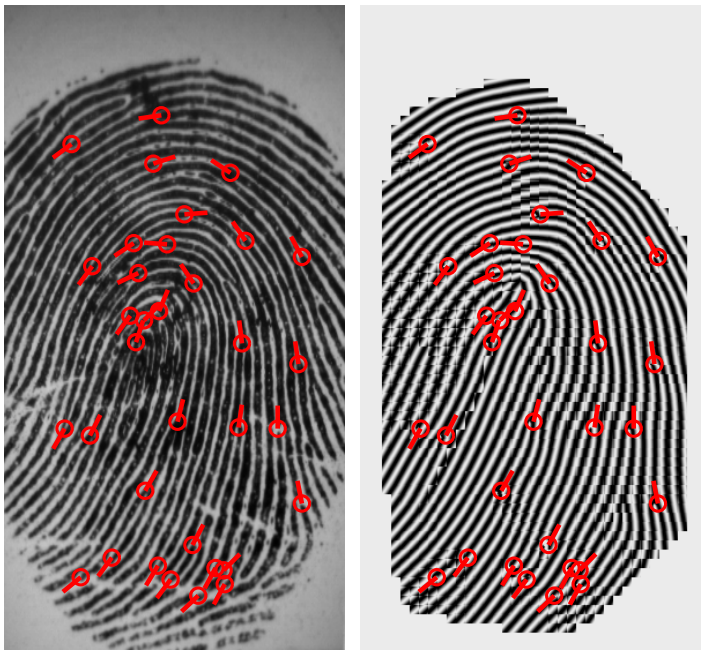
of existing biometric templates will pave a way towards the design of an "optimal" biometric template. Thus, we study a well known fingerprint representation based on minutia descriptors, namely, the binary Minutiae Cylinder Codes (MCC-B) [23], and determine its security against the recovery of the original fingerprint from which the template was obtained.

A number of techniques have been proposed that would enable an impostor to recover the biometric image from its stored template. See Table 2.1. These techniques can be broadly categorized into: i) template inversion, and ii) hill climbing. In a template inversion technique, features of the biometric image are identified from a stolen template. These features are then used to reconstruct the biometric image. In a hill climbing technique [117], however, the adversary starts with an initial guess of the biometric image which is iteratively refined based on the score obtained by matching the guessed biometric image with the stored template. Note that hill climbing techniques do not necessarily require access to the stored template but only require the match scores obtained when a reconstructed biometric image is matched with the stored template. Despite its generic nature, a disadvantage of hill climbing is that it is an iterative procedure with the number of iterations highly dependent on the characteristics of the matching algorithm. Moreover, there is no guarantee that the biometric image recovered from one system would match well with another instance of the same biometric. In the case of fingerprints, for example, a hill climbing approach may generate many spurious minutiae outside the domain of the original minutiae set or in the peripheral region of the fingerprint image. Such a reconstructed template may not lead to a high match score with another impression of the same finger. Template inversion techniques, on the other hand, do not require use of the matcher.

Fingerprints, since their inception in 1858 [56] as a method for personal identification, have been the most extensively used biometric trait. Traditionally, it was

Reference	Biometric trait	Input representation	Recovered output	Technique
Potzsch et al., 1996 [100]	Face	Elastic Bunch Graph	Face image	Template inversion
Hill, 2001 [57]	Fingerprint	Minutiae	Fingerprint image	Template inversion
Ross et al., 2007 [110]	Fingerprint	Minutiae	Fingerprint image	Template inversion
Cappelli et al., 2007 [24]	Fingerprint	Minutiae	Fingerprint image	Template inversion
Testoni and Kirovski, 2010 [125]	Iris	Iriscode	Iris	Template inversion
Feng and Jain, 2011 [42]	Fingerprint	Minutiae	Fingerprint image	Template inversion
Adler, 2003 [7]	Face	Matching System	Face image	Hill climbing
Uludag and Jain, 2004 [129]	Fingerprint	Matching System	Minutiae	Hill climbing
Yamazaki et al., 2005 [137]	Signature	Matching System	Time series data	Hill climbing
Mohanty et al., 2007 [82]	Face	Matching System	Face image	Hill climbing
Muramatsu, 2008 [86]	Signature	Matching System	Time series data	Hill climbing
Galbally et al., 2010 [50]	Face	Matching System	Face image	Hill climbing
Martinez-Diaz et al., 2011 [81]	Fingerprint	Matching System	Minutiae	Hill climbing

Table 2.1: Available techniques for recovering biometric data, given a stored template and a matching system.



(a) (b)

Figure 2.1: Fingerprint reconstruction from minutiae. (a) A fingerprint with marked minutiae, and (b) the fingerprint reconstructed from minutiae set (template) in (a), using the technique proposed in [42].

understood that it is not possible to recover a fingerprint image given its minutiae set i.e. minutiae were considered to be non-invertible. Hill [57] proposed the first template inversion technique for fingerprints. In this technique, the possible configurations of core points[1] were selected and the fingerprint orientation field was constructed for each configuration. The orientation field that best fit the given minutiae was selected and, starting from each minutia, ridge lines were then drawn along the estimated orientation field to obtain a fingerprint image. Following Hill's approach, a number of other efficient approaches have been proposed that are able to reconstruct fingerprint images from minutiae that match the original fingerprints with high accuracy. In [110], the fingerprint orientation field was reconstructed based on the direction of neighboring minutiae and the ridge lines were simulated in a manner similar to [57]. In [24], a more sophisticated technique, proposed in [132], was used to reconstruct the

[1]Core points mark the singularities in the orientation field of a fingerprint.

orientation field based on the minutiae. The orientation field estimation was followed by a filtering step that used local Gabor filters oriented along the ridge orientation to generate the fingerprint pattern. While this approach produced very realistic fingerprints compared to the earlier approaches, it also resulted in a large number of spurious minutiae. Feng and Jain [42] proposed a fingerprint image reconstruction procedure by fitting an AM-FM model to fingerprints. This approach not only generates quite realistic fingerprints, it leads to very few spurious minutiae. Figure 2.1(b) shows the reconstructed image of the original fingerprint image in Figure 2.1(a).

Given that fingerprint images can be easily and accurately recovered from minutiae based templates, it is important to find alternate fingerprint representations that are both discriminative as well as non-invertible. Minutia descriptors [59] have received significant attention as a choice for fingerprint template due to their high matching accuracy. A minutia descriptor template of a fingerprint containing n minutiae is represented as $T_{MD} = \{(x_1, y_1, \theta_1, D_1), (x_2, y_2, \theta_2, D_2), \cdots, (x_n, y_n, \theta_n, D_n)\}$, where D_i is the descriptor associated with the minutia (x_i, y_i, θ_i). The ith minutia descriptor D_i consists of discriminative features extracted in the neighborhood of the ith minutia which are invariant to rotation and translation of the fingerprint. This neighborhood information in minutia descriptors allows a robust correspondence between minutiae from multiple impressions of the same finger leading to high matching accuracy. Note that relative rotation, translation, non-rigid deformation, and small area of overlap are some of the main factors affecting the accuracy of fingerprint matching algorithms. Minutiae descriptors are, however, highly resilient to these intra-class variations. Moreover, given their high saliency, descriptor-only fingerprint templates, $T_D = \{D_1, D_2, \cdots, D_n\}$, have also been proposed that do not contain any information about the minutiae location and direction. It was shown in [23] that the descriptor-only template, T_D, leads to similar matching accuracy as the full descriptor based templates, T_{MD}, which contain descriptor as well as the location and

direction of the corresponding central minutiae[2]. The inversion of descriptor-only fingerprint templates is, however, still an open problem. Here, we show that it is indeed possible to recover the fingerprints from a descriptor-only template, T_D. Note that the techniques available in the literature to recover fingerprint image from minutiae template can also be used to recover fingerprint images from T_{MD} templates and that is why we focus our attention to descriptor-only template (T_D).

We follow a two-stage approach to reconstruct a fingerprint image from its descriptor-only template (See Figure 2.2):

1. **Local recovery**: Recovery of minutiae in the local fingerprint region associated with a minutia descriptor, D_i.

2. **Global recovery**: Linking of locally recovered minutiae sets in the first stage based on their mutual compatibilities to recover the global minutiae pattern. Once the global minutiae set is reconstructed, the fingerprint image can be obtained using any of the available techniques, e.g. [42].

We use MCC-B based descriptor-only representation to demonstrate the performance of our reconstruction algorithm. Our choice of MCC-B is motivated by the following facts: i) minutiae are not explicitly stored in the descriptor thereby making the recovery of minutiae sets and hence the fingerprint image challenging, and ii) matching accuracy achieved by MCC-B descriptor-only template represents state of the art in descriptor based fingerprint matching. The performance of the proposed reconstruction procedure is measured in terms of the match score of the reconstructed fingerprint and the original fingerprint from which the descriptor-only template was extracted using a commercial matcher.

[2]The Minutiae Cylinder Code (MCC) representation with minutiae information (T_{MD}) used in the LSA-R matcher achieved the best matching accuracy of 0.15% on FVC2006 DB2. This compared to an equal error rate of 0.33% when minutiae information was not included in the MCC representation (T_D) [23].

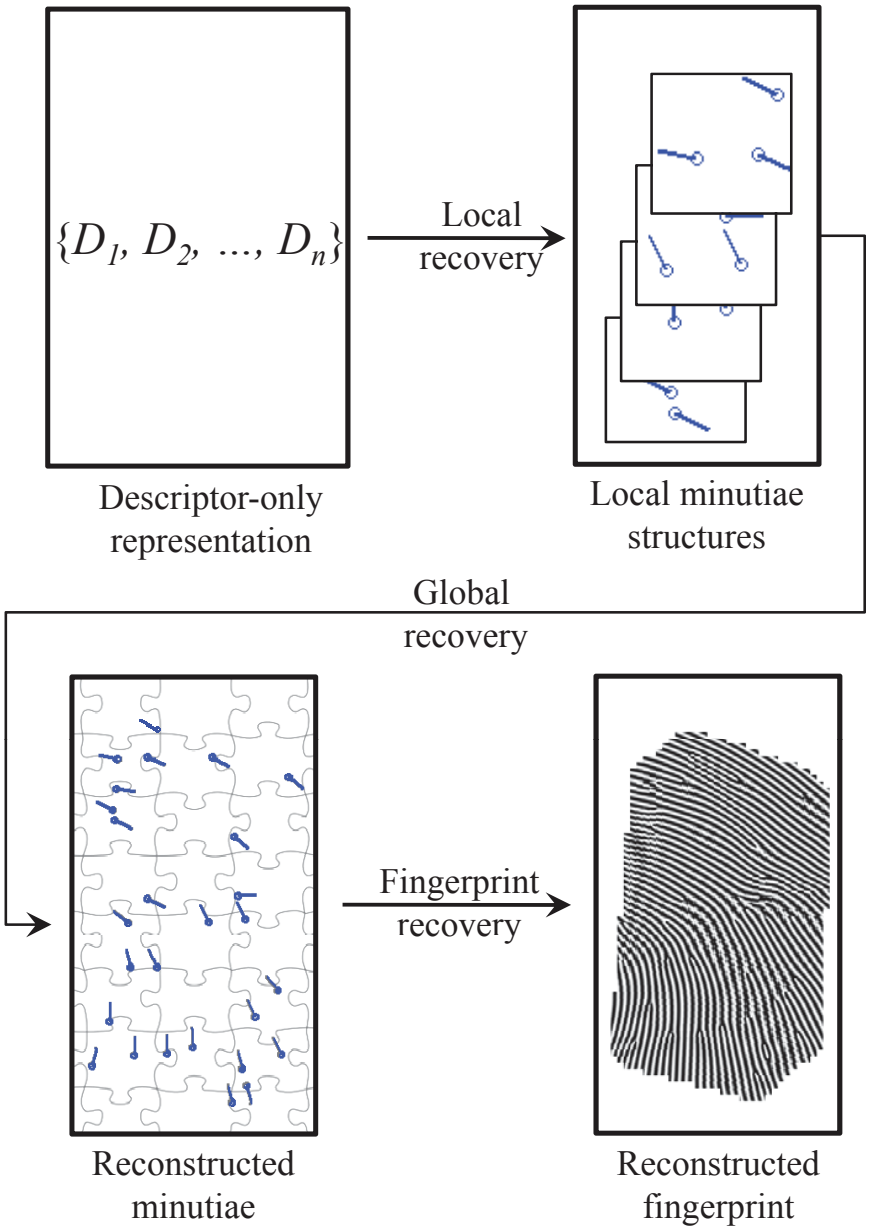


Figure 2.2: A schematic diagram depicting the various stages in recovering the fingerprint image from a descriptor-only representation (template) T_D.

Section 2.2 provides a review of the available descriptor based matching techniques. Section 2.3 provides a description of the Binary Minutiae Cylinder Codes (MCC-B). Section 2.4 provides details of the proposed reconstruction procedure. Section 2.5 details the experimental results obtained and Section 2.6 provides the summary of this chapter and suggestions for future work.

2.2 Minutia Descriptors

A descriptor associated with a minutia represents the discriminative information in the local neighborhood of the minutia. Based on the kind of information captured by a descriptor, it can be categorized into one of three main classes (see Figure 2.3): i) image features based descriptors, ii) minutiae features based descriptors, and iii) texture features based descriptors. The image features based descriptors capture the grayscale information in the local region around a minutia, the minutia features based descriptors capture the information regarding other minutiae in the local neighborhood of the central minutia, and the texture features based descriptors capture the texture related characteristics such as ridge orientation and ridge frequency. See [43] for a discussion on the characteristics and relative saliency of these three types of descriptors. Here, we are mainly concerned with the minutiae features based descriptors due to their high matching accuracy and because they appear to contain sufficient information to reconstruct the fingerprint image.

Hrechak and McHugh [59] presented the first minutia descriptor based fingerprint representation. It consisted of frequencies of eight different types of ridge based features, namely, dot, ridge ending, bifurcation, island, spur, crossover, bridge, and short ridge in a local neighborhood around each minutia in the fingerprint. These minutia descriptors were, however, mainly used for indexing fingerprints for their fast retrieval from large databases. Wahab et al. [133] extended the approach in [59] for finger-

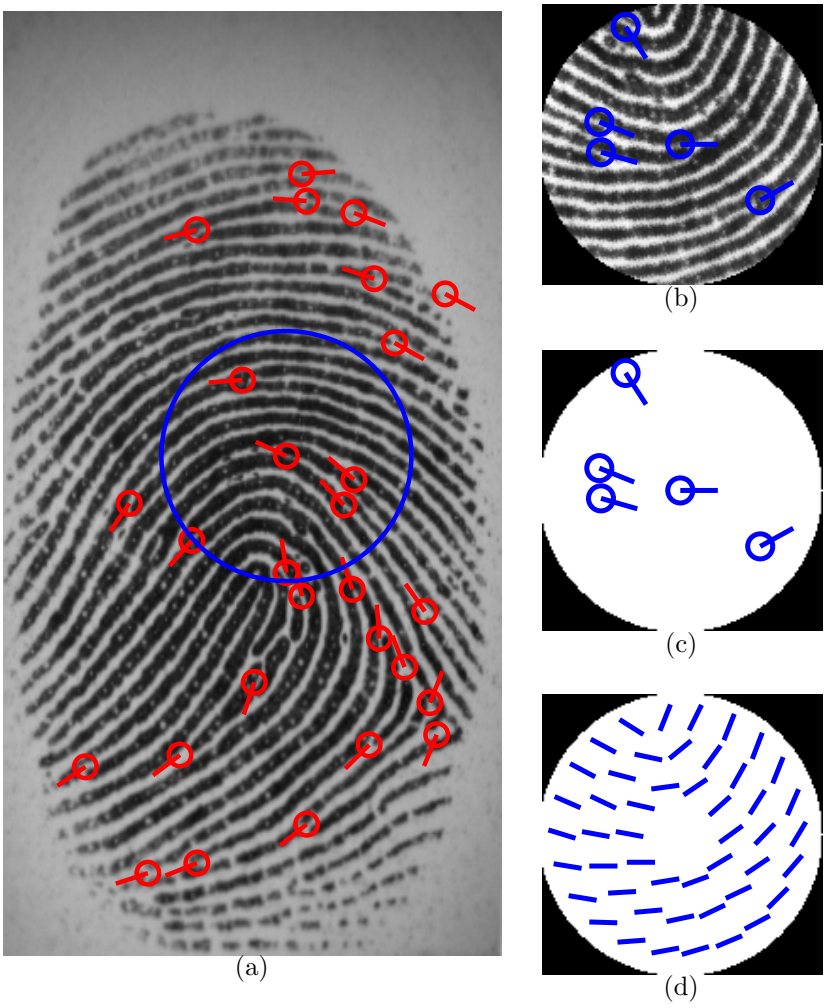


Figure 2.3: Three different kinds of descriptors: (a) fingerprint image with minutiae and local neighborhood around a minutia, (b) image features based descriptor, (c) minutiae features based descriptor, and (c) texture features based descriptor.

print matching by first matching the minutia descriptors in two fingerprints and then performing a global match of the two fingerprints based on the matched descriptors. In [133] features such as the type of the neighboring minutiae, distances to the neighboring minutiae, angles subtended by adjacent minutia in the neighborhood at the central minutia, and ridge count from a minutia to the central minutia constituted the minutia descriptors. Jiang and Yau [67] used a simplified form of the descriptor compared to the one proposed in [133], which consists of two nearest neighbors of the central minutia, but introduced a more robust matching procedure based on local

and global matching. In the local matching stage, the minutia descriptors extracted from the query were matched with the descriptors obtained from the template. In the global matching stage, the top-N (N is typically 5) most similar descriptor pairs were used to align fingerprints. A match score was computed for each such alignment and the maximum value among them was output as the final match score between the two fingerprints. This two-stage descriptor based fingerprint matching approach has become popular; a number of similar approaches with different minutia descriptors have been proposed. In [65] the minutia descriptor is essentially the same as in [67] except for a small modification in the manner in which the two neighboring minutiae are stored. In [41] a minutia descriptor that explicitly stores minutiae in the local neighborhood of the central minutia was used. The Minutiae Cylinder Code descriptor was proposed in [23] which consists of a vector indicating the presence of minutiae in various possible configurations. See Section 2.3 for further details. In [89], the descriptor, referred to as the Minutia Phase Spectrum, encodes the configuration of minutiae in a local neighborhood by computing the Fourier transform of the set of minutiae represented as Dirac-delta functions at their respective locations and binarizing the phase of the response.

Minutiae features based descriptors can be categorized into two main classes based on the manner in which the size of the neighborhood is determined:

1. **Nearest neighbor based descriptor**: Nearest neighbor based descriptors include information regarding the k-nearest neighboring minutiae of a central minutia. One advantage of these descriptors is their fixed length which allows for a fast matching. However, nearest neighbor based descriptors are not very resilient to the spurious or missing minutiae and thus lead to a lower matching accuracy. See e.g. [67, 133], and [65].

2. **Fixed radius based descriptor**: Fixed radius based descriptors include information regarding all the minutiae within a radius r of the central minutia.

Reference	Minutia descriptor	Neighborhood definition
Hrechak and McHugh, 1990 [59]	Frequency of landmarks in the neighborhood	Fixed radius
Wahab et al., 1998 [133]	k-nearest neighbors	Nearest neighbor
Jiang and Yau, 2000 [67]	2-nearest neighbors	Nearest neighbor
Tico and Kuosmanen, 2003 [126]	Ridge orientation	Fixed radius
Jea and Govindaraju, 2005 [65]	Modification of [67]	Nearest Neighbor
Feng, 2008 [41]	Ridge orientation, freq. and minutiae	Fixed radius
Feng, 2009 [61]	MinutiaCode	Fixed radius
Cappelli et al., 2010 [23]	Minutiae Cylinder Codes (MCC)	Fixed radius
Nandakumar, 2012 [89]	Minutia Phase Spectrum (MPS)	Fixed radius

Table 2.2: Various minutia descriptors available in the literature.

The main advantage of fixed radius based descriptors is that, unlike nearest neighbor based descriptors, the size of their neighborhood is not affected by the presence of missing and spurious minutiae. However, the number of minutiae present in the neighborhood associated with a descriptor may vary across multiple impressions of the fingerprint due to the presence of missing or spurious minutiae. This leads to a computationally demanding matching algorithm. See e.g. [28, 59, 105]. The matching speed, however, can be improved by extracting fixed length aggregate features from the neighboring minutiae for matching purpose. See e.g. [23, 89].

Table 2.2 lists various minutia descriptors available in the literature.

In addition to the use of minutia descriptors in the local-global matching procedure, aggregates of descriptors have also been effectively used in obtaining a fixed length representation of a fingerprint. In [21], e.g., the descriptors explicitly containing neighboring minutiae are extracted from a fingerprint and are matched with a database of minutiae sets. The obtained match scores are thresholded to form a

binary feature vector to be used as a fingerprint template. Privacy preserving modifications have also been applied to minutia descriptors by transforming the descriptors according to certain user specific information. For example, in [138], a user password is used to transform the set of minutia descriptors to obtain a secure representation. Given this extensive and growing body of work on minutia descriptors, it is imperative that we analyze various aspects related to minutia descriptor's security and matching accuracy.

2.3 Binary Minutiae Cylinder Codes

Minutia Cylinder Codes (MCC) were proposed as minutia descriptors for fingerprint matching in [23]. An MCC consists of fixed length vectors associated with the minutiae present in a fingerprint. These vectors, called the cylinder codes, represent various possible configurations of a minutia present in the local neighborhood of the central minutia. A mapping

$$\mathcal{H} : C \mapsto X \tag{2.1}$$

is intuitively defined from the set C of cells associated with the cylinder code to the set X of possible configurations of a neighboring minutia represented by a 3-tuple (x, y, θ). A minutia (x, y, θ) belong to the neighborhood of (x_c, y_c, θ_c) if

$$\sqrt{(x - x_c)^2 + (y - y_c)^2} < r \tag{2.2}$$

where r is the radius of the descriptor neighborhood. This restriction leads to a cylindrical shape of the descriptor in x, y, and θ coordinates. The value associated with a cylinder cell is computed as the probability of finding a minutia, at a given location around the central minutia, in another impression of the same finger. This probability is computed under the assumption that the differences in location and direction

of two corresponding minutiae in different impressions of the same fingerprint have a Gaussian distribution. Figure 2.4 graphically depicts a minutia cylinder and the corresponding minutia cylinder code. Note that the MCC descriptors combine the advantages of both the nearest neighbor as well as fixed radius based descriptors. Not only is the matching based on MCC based descriptors fast due to its fixed length, it is also robust to missing and spurious minutiae, leading to high matching accuracy. Further, construction of MCC does not require any additional information besides minutiae.

A limitation of an MCC based descriptor in its original form is its large size. However, Cappelli, et al. [23] showed that most of the discriminative information in the descriptor is retained even if each element of the cylinder code is quantized to a single bit. The resulting representation is referred to as the bit based or binary MCC (MCC-B). Here, we consider the MCC-B representation instead of the basic MCC due to its relatively compact nature. The proposed inversion scheme, however, is also applicable to MCC as well. In the specific construction of MCC we are considering, the local region around the central minutia is tessellated into a rectangular spatial grid having 16×16 elements and the minutia direction is quantized into 5 bins leading to a cylinder code with $1,280$ cells. The validity of each cell, i.e. whether the cell is in the foreground region of a fingerprint image and it satisfies the restriction posed in Eq. (2.2), is encoded as bits associated with each of the 16×16 grid points. With these parameters, a typical fingerprint consisting of 40 minutia would require $\sim$ 8 kilobyte (kB) of storage which is an order of magnitude smaller than the storage requirement of the basic MCC template which is $\sim$ 200kB assuming the use of a 32-bit floating point number. The relatively compact and bit based MCC-B template also makes it suitable for efficient fingerprint template storage and matching, and for higher security in small/low-end devices such as smart cards. Figure 2.5 depicts the MCC-B template for a typical minutia neighborhood.

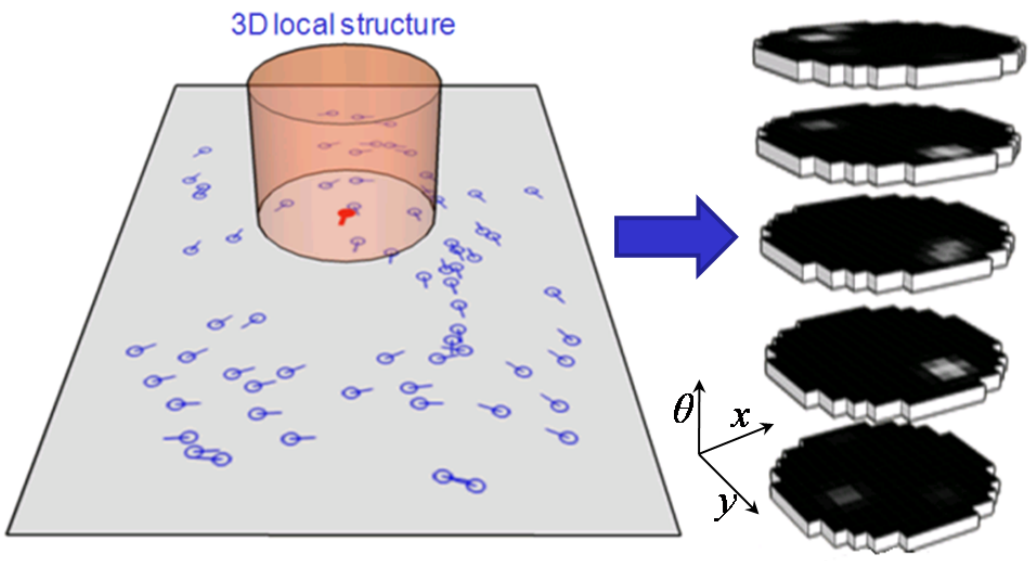


Figure 2.4: A cylinder associated with a minutia and the corresponding MCC descriptor. Each of the six discs on the right represent the cells of the cylinder corresponding to the six different minutiae directions. Source: http://biolab.csr.unibo.it/ResearchPages/graphics/MCC1.png

2.4 Reconstruction From Descriptors

In this section we describe the proposed procedure that is able to reconstruct the fingerprint image from an MCC-B descriptor-only template. The proposed fingerprint reconstruction procedure involves two stages: i) local minutiae recovery and ii) global minutiae recovery.

2.4.1 Local Minutiae Recovery

The local minutiae recovery stage involves the procedure to recover the minutiae from individual descriptors. For success in this stage, it is necessary that the descriptors contain sufficient information regarding the minutiae in the neighborhood of the central minutia. This requirement is adequately satisfied by the MCC-B descriptors and the following procedure is used to recover the neighborhood minutiae from them.

Given foreground bits (i.e. bits with value 1) in the binary cylinder code associated with a minutia, we obtain the corresponding minutiae coordinates (x, y, θ) based on

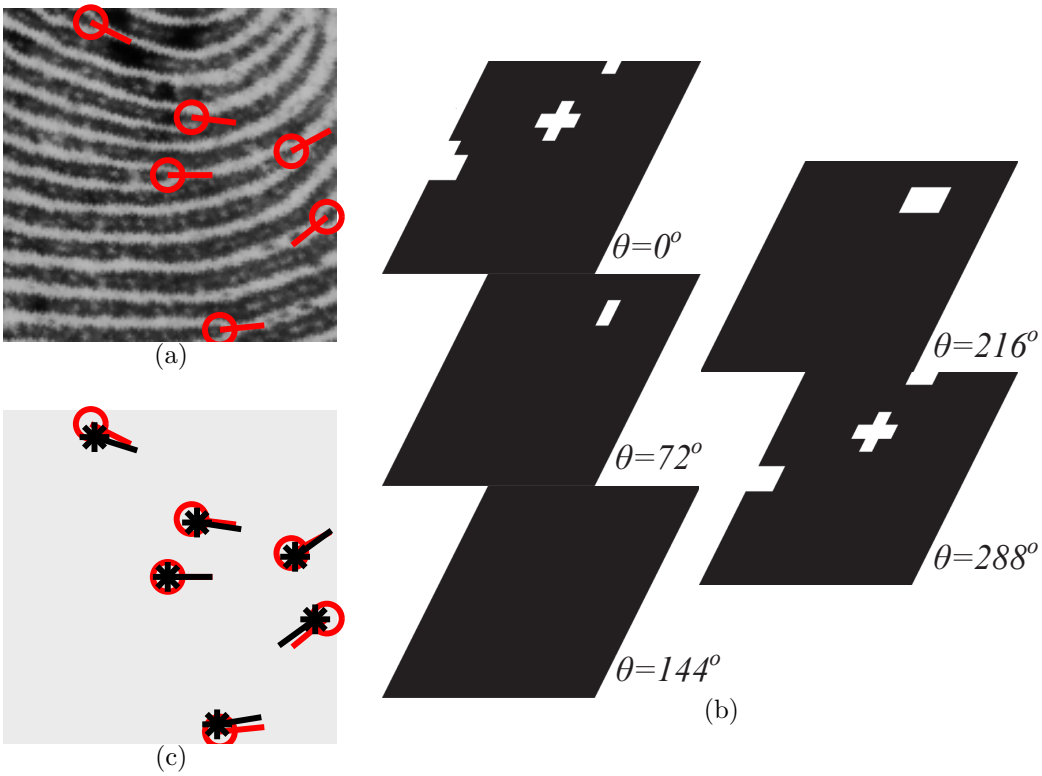


Figure 2.5: Local minutiae reconstruction: (a) original minutiae in a local region of a fingerprint, (b) associated bit-planes of the MCC-B for the five different equally separated minutiae directions, and (c) minutiae reconstructed from MCC-B descriptor (in black) overlaid on original minutiae (in red). Note that the white regions in bit-planes correspond to the neighboring minutiae and the plane in which the white regions appears depends on the direction of the corresponding minutia. Here, each plane is a 16×16 block representing a region of size 150×150 in the fingerprint image.

the mapping defined in Eq. (2.1). Since a single minutia in the original fingerprint can lead to multiple foreground bits, and each foreground bit may receive contributions from multiple minutiae, we cluster all the available foreground bits using an average-link hierarchical clustering algorithm (also known as Unweighted Pair Group Method with Arithmetic Mean (UPGMA)) [60] with a distance based tree pruning criteria. The specific measure of distance between clusters, i.e. average-link, is used because of its relevance with respect to spherical clusters. In order to use the MATLAB routines for UPGMA tuned for euclidean distance, we represent a minutia as a 4-tuple $(x, y, 10 * sin(\theta), 10 * cos(\theta))$. The tree pruning criteria is set to 20 in our

experiments. The resulting cluster centers, after converting them back to the (x, y, θ) representation, are considered as the set of recovered minutiae. Figure 2.5 shows the minutiae reconstructed from a cylinder code. Note that the minutiae are in general successfully recovered despite the quantized (binary) nature of the MCC-B code. Most of the errors in minutiae recovery in terms of localization and presence of missing or spurious minutiae can be attributed to relatively small number of foreground bits associated with the boundary minutiae and presence of minutiae outside the boundary of the local region in close vicinity. Based on our experiments on FVC 2002 fingerprint database-2 [79], the average localization error in the minutiae is 2.5 pixels, and the average deviation in the minutiae direction is 2°. On average, there are 0.18 missing minutiae and 0.61 spurious minutiae in a recovered local minutiae set. In this analysis, an original minutia $u_o = (x_o, y_o, \theta_o)$ is considered missing if there is no recovered minutia $u_r = (x_r, y_r, \theta_r)$ such that $d(u_o, u_r) < 15$, where

$$d(u_o, u_r) = d_{xy}(u_o, u_r) + 0.2 * d_{\theta}(u_o, u_r) \tag{2.3}$$

and

$$d_{xy}(u_o, u_r) = \sqrt{(x_o - x_r)^2 + (y_o - y_r)^2} \tag{2.4}$$

$$d_{\theta}(u_o, u_r) = (min(|\theta_o - \theta_r|, 360 - |\theta_o - \theta_r|)). \tag{2.5}$$

2.4.2 Global Minutiae Recovery

The global minutiae recovery stage involves linking the locally recovered minutiae sets in order to obtain a global minutia pattern. This stage consists of two main phases. In the first phase, referred to as the link assessment phase, compatibilities between pairs of local minutiae sets are computed for all possible ways in which two local sets can be linked. In the second phase, referred to as the link aggregation phase, the

local minutiae sets are selected based on their compatibility scores and then linked together in order to recover the global minutia pattern.

Link assessment

Consider two recovered local minutiae sets, $X_p = \{u^1, u^2, \cdots, u^{k_p}\}$ and $X_q = \{v^1, v^2, \cdots, v^{k_q}\}$. We align each minutia in X_p with each minutia in X_q separately and compute the compatibility scores of all the $k_p \times k_q$ possible links. Note that a specific alignment between two local minutiae sets is referred to as a link. See Figure 2.6. A link is represented as a 4-tuple (i, j, p, q) which indicates that the ith minutia in the pth local minutiae set overlaps with the jth minutia in the qth local minutiae set. The compatibility between two local minutiae sets aligned according to a link is determined by two main factors: (i) number of overlapping minutiae between the two local sets, and (ii) number of minutiae in one of the local sets that do not belong to the region associated with the other local set. Note that the former factor indicates the validity of the link whereas the later indicates the capacity of the link to increase the total number of minutiae recovered.

A link is considered valid if the two minutiae being matched according to that link are accurate recoveries of the same original minutiae i.e. if the distance between the recovered minutia and the original minutia is less than certain threshold. Distance between two minutiae $u_1 = (x_1, y_1, \theta_1)$ and $u_2 = (x_2, y_2, \theta_2)$ is computed using Eq. (2.3). The compatibility score s_{ij}^{pq}, when the ith minutia in the pth set is aligned with the jth minutia in the qth set, is given by

$$s_{ij}^{pq} = \sum_{t=1:k_p} S(u_t, X_q') + \sum_{t=1:k_q} S(v_t, X_p') + w(\alpha_p + \alpha_q) \tag{2.6}$$

where

$$S(u, X) = \max_{v \in X} \frac{1 - d_\theta(u, v) * \pi/180}{d_{xy}(u, v) + 5} \tag{2.7}$$

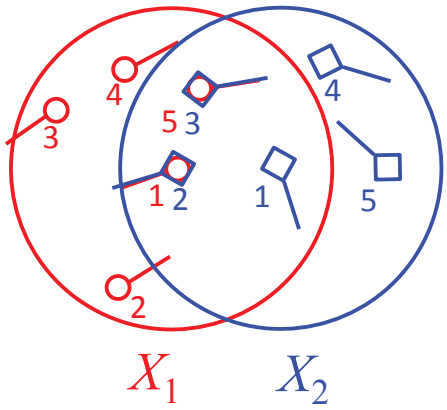


Figure 2.6: Depiction of two links $(5, 3, 1, 2)$ and $(1, 2, 1, 2)$ between two minutiae sets X_1 (represented in red) and X_2 (represented in blue) where the 5th and 1st minutiae in X_1 are overlaid on the 3rd and 2nd minutiae in X_2, respectively. Note that a link (i, j, p, q) indicates that the ith minutia in the pth local minutiae set overlaps with the jth minutia in the qth local minutiae set.

d_{xy} and d_θ are defined in Eqs. (2.4) and (2.5), and the sets X'_p and X'_q are the aligned minutiae sets corresponding to X_p and X_q, respectively.

The values of α_p and α_q are computed as

$$\alpha_p = |X_p| \text{ where } X_p = \{u_i | u_i \in X_p, d_{xy}(u_i, c_q) > t\} \tag{2.8}$$

$$\alpha_q = |X_q| \text{ where } X_q = \{u_i | u_i \in X_q, d_{xy}(u_i, c_p) > t\} \tag{2.9}$$

where c_p (c_q) is the central minutia in X_p (X_q), and t denotes the radius of the local neighborhood. The parameter t is set to 75 which is the default value for the radius of the neighborhood in the SDK for extracting the MCC-B descriptors provided by the University of Bologna [23]. The weight w determines the trade-off between the effect of number of overlapping minutiae among the two local sets being linked and the effect of number of minutiae in the non-overlapping region of the two local neighborhoods on the link compatibility score. Its value is set to 0.05 in our experiments.

The set of links determined between two local minutiae sets using the above procedure has some redundancy. Consider two local minutiae sets X_p and X_q as defined

above. When a link $l_i = (g, h, p, q)$ with compatibility score s is "executed", i.e. when the gth minutia of the X_p is overlaid on the hth minutia of X_q, a set of other minutiae from the two local sets may also have mutual distances (see Eq. (2.3)) smaller than a specified threshold. Figure 2.6 depicts a merged minutia set obtained by executing a link; the same configuration of merged minutia set is also obtained while executing another link as noted in the Figure 2.6. Let the pairs of minutiae in these sets correspond to links $\{l_{i_1}, l_{i_2}, ..., l_{i_k}\}$ with compatibility scores $\{s_{i_1}, s_{i_2}, ..., s_{i_k}\}$. We note that all the links $\{l_{i_1}, l_{i_2}, ..., l_{i_k}\}$ when separately executed lead to similar merged minutiae set and thus only the link with the largest compatibility score among them is retained and the remaining links are discarded. This procedure is performed for all the links associated with each pair of local minutiae sets. A description of this procedure is provided in Algorithm 2.1.

Algorithm 2.1 Redundant links removal algorithm

Input: X_p, X_q : Two local minutiae sets being linked
Output: L: Retained links
Procedure $L = LinkRed(X_p, X_q)$

Let l_i : ith link among a set of $|X_p| \times |X_q|$ possible links
Let red: redundancy indicator; $red(i) = 1$ if the ith link is redundant, otherwise $red(i) = 0$
for $i = 1$ to $|X_p| \times |X_q|$ **do**
    $red(i) = 0$
end for
for $i = 1$ to $|X_p| \times |X_q|$ **do**
    Let $\{l_{i_1}, l_{i_2}, ..., l_{i_k}\}$ = links corresponding to overlapping minutiae obtained by executing l_i {Note: $l_i \in l_{i_1}, l_{i_2}, ..., l_{i_k}$}}
    Let $s_i = s_{gh}^{pq}$ where $l_i = (g, h, p, q)$
    Let $t_{ret} = \arg\max_k \{s_{i_k}\}$
    for $t = 1$ to k **do**
        if $t \neq t_{ret}$ **then**
            $red(i_t) = 1$
        end if
    end for
end for
$L = \{l_i | red(i) = 0\}$

Link aggregation

This stage involves merging of locally recovered minutiae to obtain a global minutiae pattern. First, a set of links is selected from the available links based on their compatibility scores as described in Section 2.4.3. Note that executing this set of links, or in other words merging the locally recovered minutiae sets according to these links, may lead to disjoint clusters of merged minutiae sets. Consider a set of links $\{(1,1,1,2),(1,1,2,3),(1,1,5,6)\}$. Executing this set will lead to two disjoint minutiae sets, where the larger set is obtained by merging local minutiae sets $1,2$ and 3 while the smaller set contains minutiae from local sets 5 and 6. Given a set of links, we identify the largest cluster prior to executing the links and execute only those links that are associated with the largest cluster.

Given a set L of selected links, a single link hierarchical clustering algorithm [60] over the sets of locally recovered minutiae sets is used to identify the clusters. For the purpose of this clustering, the distance between two locally recovered minutiae sets X_p and X_q is measured as

$$d(X_p, X_q) = \begin{cases} 0 & \text{if } (i,j,p,q) \in L \\ 1 & \text{otherwise} \end{cases} \tag{2.10}$$

Algorithm 2.2 formally describes this procedure for selecting the links belonging to the largest cluster.

Once the links associated with the largest cluster of local minutiae are selected, the next step is to merge these links as described in Algorithm 2.3. Here, among the links associated with the largest cluster, we start with the first link (with the highest similarity) and merge the two local minutiae sets associated with the link in the manner determined by the link. In order to merge the minutiae sets indicated in a link (i,j,p,q), the minutiae in X_q are transformed such that the jth minutia in X_q

Algorithm 2.2 Largest cluster selection algorithm

Input: L : Available set of links
Output: L_s: Links in largest cluster
Procedure $L_s = LargestLinkCluster(L)$

1: Let X_i be the ith local set
2: Let

$$\{D\}_{pq} = \begin{cases} 0 & \text{if } (i,j,p,q) \in L \\ 1 & \text{otherwise} \end{cases}$$

3: C = SingleLink(D) {Note: The SingleLink procedure outputs a set of clusters. Each cluster is a sets of indices into the data points belonging to that cluster}
4: $L_s = \phi$
5: $l = \arg\max_i |C\{i\}|$
6: **for** $i = 1$ to $|L|$ **do**
7: Let $L\{i\} = (g,h,p,q)$
8: **if** $p \in C\{l\}$ OR $q \in C\{l\}$ **then**
9: $L_s = L_s \cup L\{i\}$
10: **end if**
11: **end for**

overlaps with the ith minutia in X_p. The transformed minutiae set X'_q obtained from X_q is overlaid on X_p to obtain the matching minutiae points between the two sets. The alignment is then adjusted based on these matching points using the registration procedure described in [15]. Since the registration procedure in [15] requires sets of two dimensional points, minutiae direction is incorporated during registration by temporarily placing points at a distance of 20 pixels from each minutia along its direction. The transformed minutiae belonging to set X_q are then added to the set X_p to obtain the merged minutiae set. This procedure is referred to as $Merge$ in Algorithm 2.3.

After merging the two local minutiae sets based on the first link, we visit the next selected link in order. If this link has one associated local minutiae set that is already considered in the merged minutiae set, the second minutiae set is then combined with the merged set according to the link. Once all the selected links are considered, the links that have not been incorporated in the merged set are consider again in order.

This procedure is repeated till all the links associated with the largest cluster are executed.

Due to merging of local structures, there are overlapping minutiae in the recovered global minutiae set. We thus cluster all the minutiae using an average-link hierarchical clustering algorithm as described in Section 2.4.1. The centers of the clusters obtained as a result of average-link clustering are considered as the final set of globally recovered minutiae. The recovered global configuration of minutiae is finally transformed into a fingerprint image using the procedure described in [42]. In order to avoid inordinately large execution time we do not reconstruct fingerprint images in cases where the number of detected singular points is greater than 8. Further, we generate eight different rotations of the recovered fingerprint image so that at least one of the rotated images has a nearly upright position. Note that certain fingerprint matchers do not match two fingerprints from the same finger if the relative rotation between the fingerprints is greater than certain threshold. Figure 2.7 shows an example reconstruction of a fingerprint when only the top-20 links with the highest compatibility scores, irrespective of being valid or invalid, are used to recover the global minutiae pattern. In Figure 2.7, we first show the 12 locally recovered minutiae sets having largest number of minutiae. Then we show the indices of the constituent local minutiae sets associated with 20 most compatible links in order. In this case all the top-20 links were valid links. Note that in the links having large compatibility scores belong to minutiae sets having large number of minutiae. Next we show the snapshots of the reconstructed minutiae sets during first, middle and final execution of a link. The last local minutiae set merged is shown in blue and the corresponding link is marked on the image. The order of link execution is determined by the procedure described in Section 2.4.2. Note that the link marked $4-1$ was not executed in the second place despite having the second highest compatibility score since the first link i.e. the link marked $2-3$ did not involve either the first or the

fourth minutia set. We then show the minutiae obtained after clustering the merged minutiae set obtained by executing all the links. This minutia set is marked as "Final Result" in the figure. We then show the reconstructed fingerprint generated based on this reconstructed minutia set. This reconstructed fingerprint reasonably captures a significant portion of the complete fingerprint leading to a high match score.

Algorithm 2.3 Global minutiae recovery algorithm

Input: X: Set of recovered local minutiae sets
Output: X_g : Set of globally recovered minutiae
Procedure $X_g = GlobalRec(X)$

Let $L = \phi$
for $(X_p, X_q) \in X$ **do**
  $L = L \cup LinkRed(X_p, X_q)$ {Note: See Algorithm 2.1}
end for
Let L' = links selected among L based on procedure defined in 2.4.3
Let $L_s = LargestLinkCluster(L)$
{Note: Algorithm 2.2}
$X_m = \phi$
loop
  for $l = (i, j, p, q) \in L_s$ **do**
    switch (X_m)
      case $X_M = \phi$**:**
      X_m=Merge(X_p, X_q, l)
      case only X_p is already merged with X_m**:**
      X_m=Merge(X_m, X_q, l)
      case only X_q is already merged with X_m**:**
      X_m=Merge(X_m, X_p, l)
    end switch{Note: The procedure Merge(X_p, X_q, l) outputs the minutia set when the sets X_p and X_q are merged according to the link l. The specific details are provided in Section 2.4.2.}
  end for
  if all X_p associated with L_s are merged in X_m **then**
    break;
  end if
end loop
X_g = cluster centers of X_m {Note: Average link hierarchical clustering is applied on X_m. See Section 2.4.2 for details.}

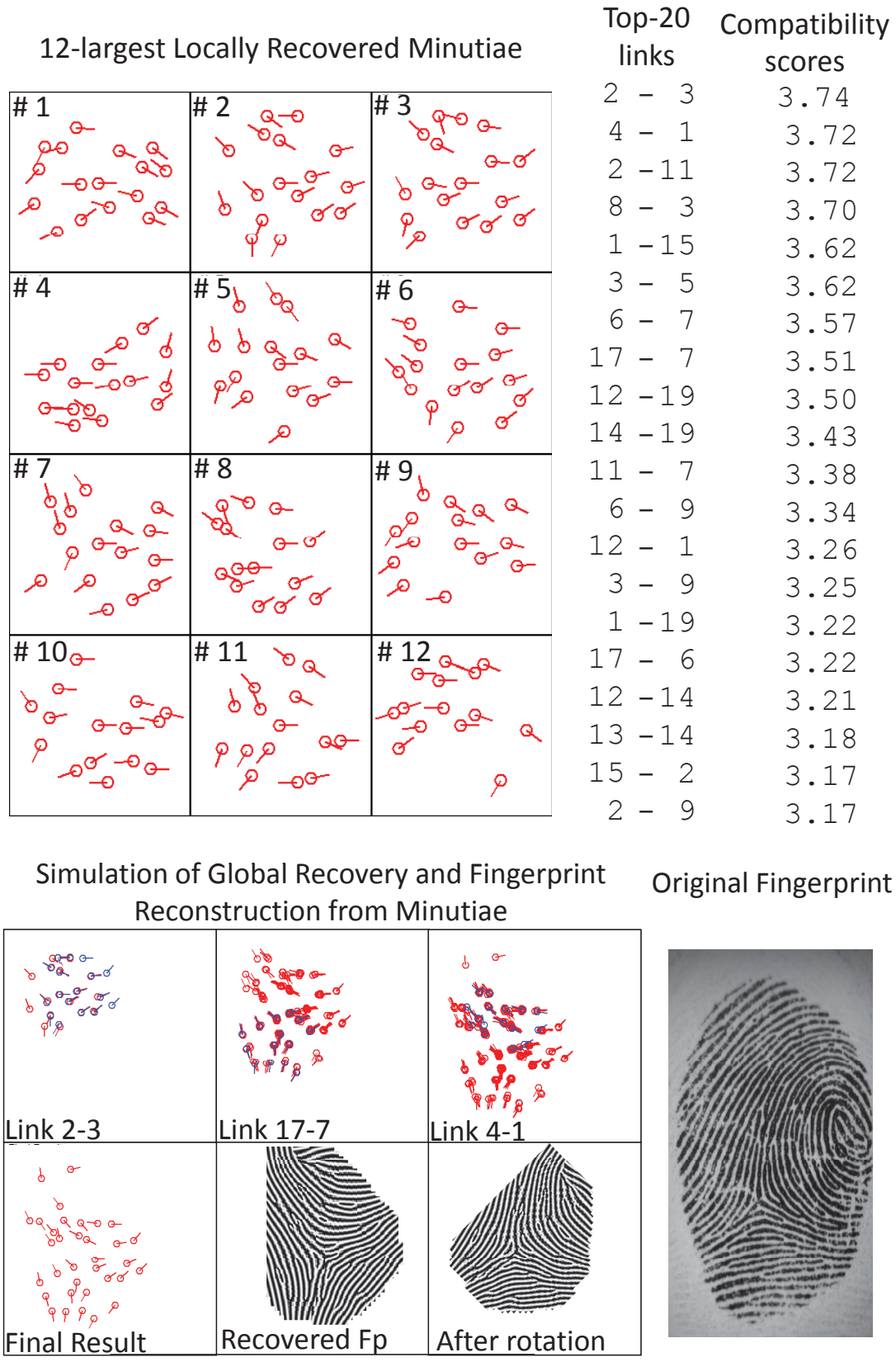


Figure 2.7: Simulation of the fingerprint reconstruction procedure. See Section 2.4.2 for a description of this figure.

2.4.3 Link Selection and Complexity Analysis

During the global recovery stage, even after the redundant links have been removed using Algorithm 2.1, a large number of invalid links still remain in the selected set of links. Thus, it is important to select a set of links that have a large probability of being valid and use them to recover the global minutiae pattern as in Algorithm 2.3. An intuitive way to select a set of links among the possible links is by taking a fixed number of the most compatible links, where the compatibility of a link is measured using the procedure described in Section 2.4.2. The only parameter in this procedure is the number of links considered. It is expected that as the number of links considered is increased, the reconstruction accuracy will increase, but only till a certain extent. Considering more links can also deteriorate the reconstruction accuracy due to execution of a large number of invalid links during the reconstruction procedure. See Figure 2.9 for the trend in reconstruction accuracy as the number of links considered is increased. To improve the reconstruction accuracy beyond the accuracy that can be achieved by considering only the top few most compatible links, it is imperative to identify and discard the invalid links from the chosen subset of links. One way to achieve this is by performing multiple trials of reconstruction, where in each trial, a different subset of links is discarded. We use the strategy described in Algorithm 2.4 for selecting the desired links for removal.

Given a maximum number n of links considered (1000 in our experiments), we simultaneously perform the following procedure for each value of $k = 1$ to n. Given top-k links, all possible subsets of links of size $i = 1, 2, ...k$ are incrementally selected and discarded, and the remaining links are used to reconstruct the fingerprint. If we limit the number of candidate reconstructions allowed to th_C, the value of k is limited

by the following condition:

$$\sum_{i=1}^{min(k-d,d)} \binom{k}{i} < th_c \tag{2.11}$$

where d is the number of invalid links among the first k links. Note that the number of locally recovered minutiae sets merged to obtain the global minutiae pattern depends on the frequency of invalid links among the top few links with the largest compatibility scores. A template for which there are very few invalid links with high compatibility scores will allow a large number of locally recovered minutiae sets to be merged thereby leading to a global minutiae pattern with larger number of minutiae. However, if there are large number of invalid links among the top few most compatible links, a large number of candidate minutiae sets would need to be generated in order to recover a small number of true minutiae. For example, if there are two invalid links present among the top 20 valid links, 210 candidate minutiae sets need to be generated in order to obtain a global minutiae set that involves all the 18 correct links.

Algorithm 2.4 Link selection algorithm

Input: $L = \{l_1, l_2, ..., l_n\}$: Ordered set of links, n: Number of most compatible links considered, th_c: Maximum number of trials allowed

Output: $\mathbb{L}$: A list of sets of links to be considered for reconstructing global minutiae pattern

Procedure $\mathbb{L}$ =LinkSelection(L, n, th_c)

1: **forall** $^3k = 1$ to n **do**
2: $C = 0$
3: **for** $t = 1$ to k **do**
4: Let $L_t = \{l_1, l_2, ..., l_k\}$
5: Let G is randomly chosen s.t. $G \subset L_t$ AND $|G| = t$
6: $L_t = L_t \backslash G$
7: $\mathbb{L} = \mathbb{L} \cup L_t$
8: $C+ = \binom{k}{t}$
9: **if** $C > th_c$ **then**
10: **break;**
11: **end if**
12: **end for**
13: **end forall**

2.5 Experiments

We used the FVC 2002 DB2 fingerprint database [79] in our experiments and analysis which contains 100 different fingers with 8 impressions per finger. The fingerprint images are of size 296×560 and they were captured at a resolution of 569 ppi. First, the MCC-B descriptors (with neighborhood of radius 75 pixels tessellated into a 16×16 grid and minutia direction quantized into five bins) were extracted from the fingerprints using the SDK provided by the University of Bologna [23]. Given these descriptors, our goal is to reconstruct the fingerprints following the proposed approach.

The reconstruction procedure was tested under two main scenarios:

1. Same impression scenario: How similar is the reconstructed fingerprint image to the original fingerprint image from which the MCC-B template was obtained?

2. Different impression scenario: How similar is the reconstructed fingerprint image to a different impression of the same finger from which the descriptors were obtained?

The success of fingerprint reconstruction under these two scenarios is presented in the form of Receiver Operating Characteristic (ROC) Curves of the Morpho fingerprint matcher [85]. Note that similar scenarios have also been considered for evaluation of the fingerprint reconstructed from minutiae in [42]. For the same impression scenario, the corresponding genuine match scores are obtained by matching the reconstructed fingerprint with the original fingerprint from which the template was derived. For the different impression scenario, the reconstructed fingerprint is matched with a different impression of the finger to obtain the genuine match score. This leads to $5,600$ genuine match scores for each scenario. The impostor scores are

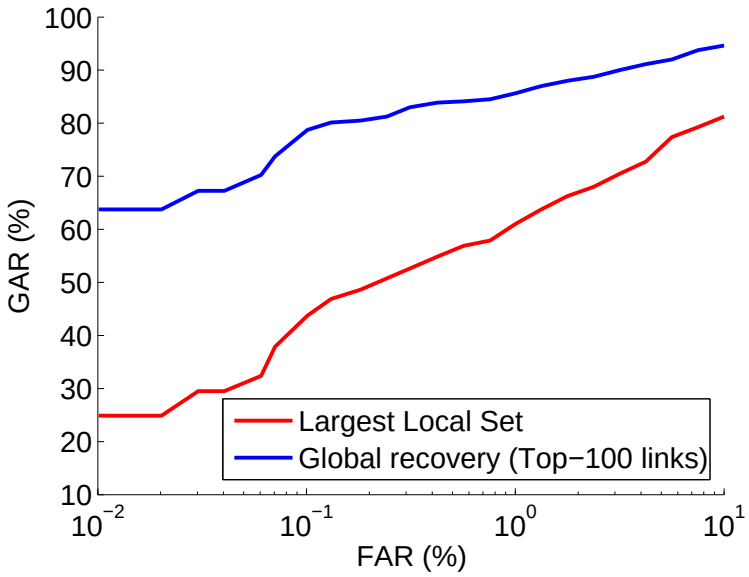
[3] **forall** is the parallel for-loop which runs all the instances of the loop simultaneously.

obtained by matching the first impression of each finger with the first impressions of all other non-mate fingers. This leads to $9,900$ impostor match scores for both the scenarios. Note that while matching a reconstructed fingerprint with a stored template, the maximum match score obtained from each of the eight rotated versions of the reconstructed fingerprints is used as the final match score.

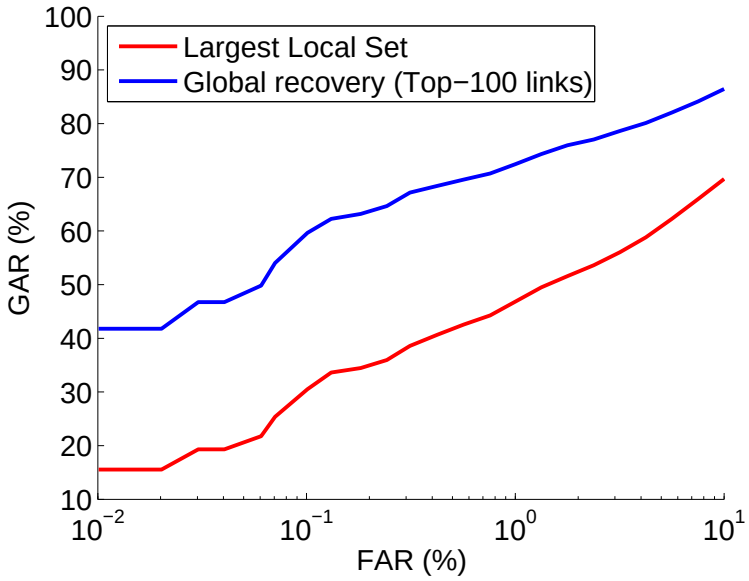
In the first experiment, we evaluate the scenario where the adversary recovers the local minutiae set from each descriptor in a template and uses the locally recovered minutiae set with the largest number of minutiae as the global minutiae set to recover the original fingerprint image. In approximately 25% of the $5,600$ genuine match cases, the reconstructed fingerprints, when matched with same original fingerprint from which the template was derived, were accepted by the fingerprint matcher operating at an FAR of 0.01%. However, when the reconstructed fingerprint was matched with a different impression of the same finger, only 16% of the cases were accepted. The corresponding ROC curves are shown in Figure 2.8.

Next, we evaluated the matching accuracy obtained when the fingerprints were reconstructed by linking the top-k most compatible links without considering their validity. It was observed that the reconstruction accuracy first improves as the number of links considered is increased but if the number of links is increased beyond certain limit (100 here), the reconstruction accuracy declines likely due to execution of invalid links. This trend is shown in Figure 2.9 where the genuine accept rate of the reconstructed fingerprints is plotted against the number of links considered at an FAR of 0.01%. As can be observed in Figure 2.9, the reconstruction accuracy peaks at case when top-100 links are considered. The ROC curve corresponding to this case is also shown in Figure 2.8.

Note that the adversary may also try to select the templates from which it is easier to reconstruct the fingerprint images. We evaluated this scenario by using the number of minutia descriptors in the template as a criteria to select the templates for

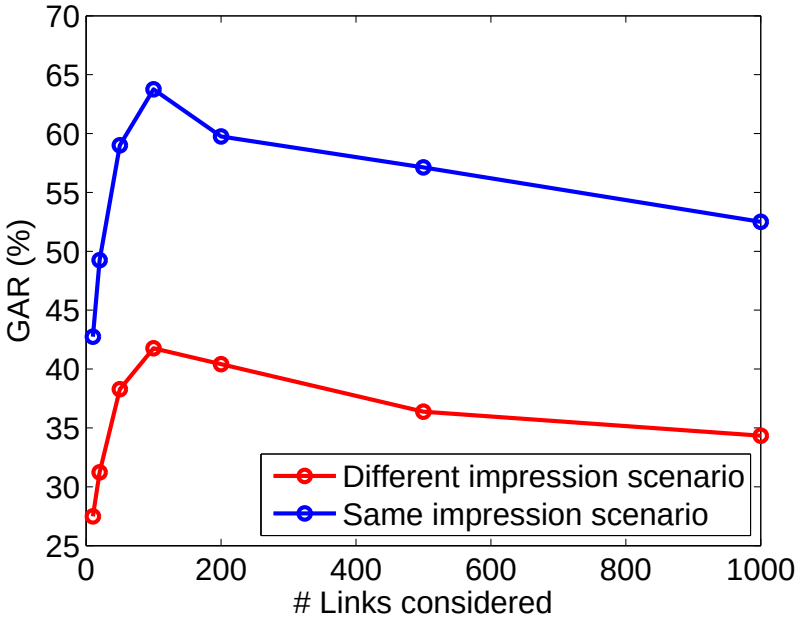


(a)

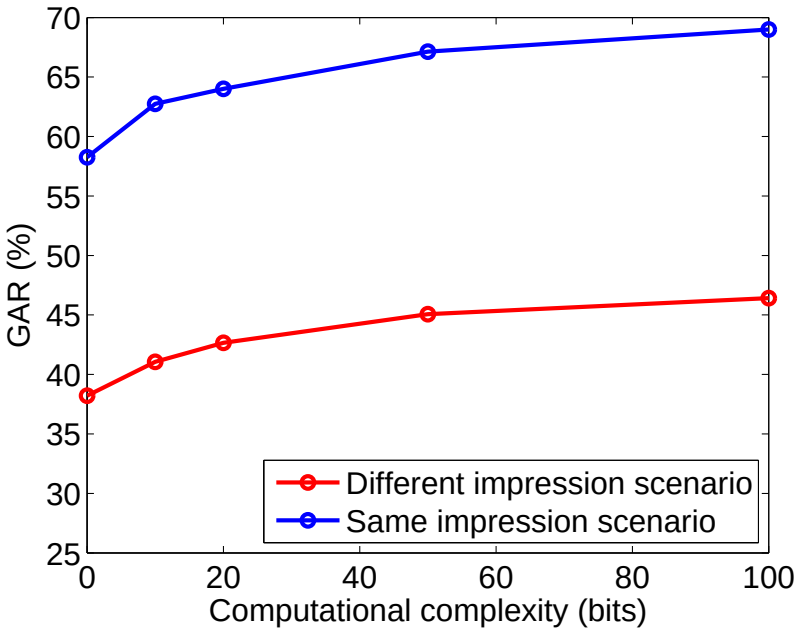


(b)

Figure 2.8: ROC curves for cases when (a) the reconstructed fingerprints are matched with the corresponding original fingerprints from which the templates were derived, and (b) the reconstructed fingerprints are matched with a different impression of the same finger.



(a)



(b)

Figure 2.9: Relationship between the various link selection criteria and genuine accept rate (GAR) at an FAR of 0.01%. (a) Effect of increasing the number of links considered without checking their validity on the matching accuracy, and (b) effect of increasing the computational complexity on matching accuracy.

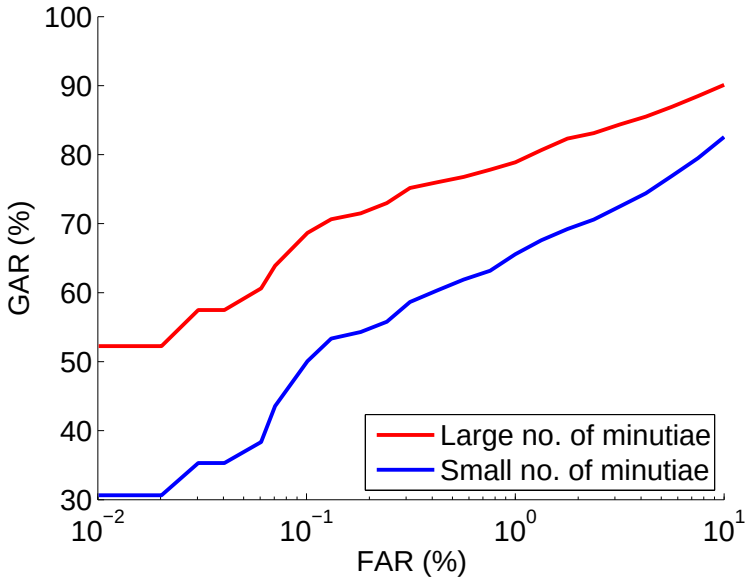


Figure 2.10: ROC curves corresponding to the case when the reconstructed fingerprints are divided based on the number of minutiae descriptors in the template. Here the top-100 most compatible links are executed for reconstructing the fingerprint. The threshold on the number of minutiae used to categorize the reconstructed fingerprints into "Large no. of minutiae" and "Small no. of minutiae" is 34.

reconstruction. If only the templates that have a minimum of 34 minutiae, which is the median value of the number of minutiae in a fingerprint in the database considered, are used for reconstruction, the genuine accept rate at an FAR of 0.01% is increased to 52% from 42% in the different impression scenario. The ROC curves for templates having number of minutiae more than or less than 34 are shown in Figure 2.10.

Next, we analyze the recognition accuracy obtained when a large number of candidate reconstructed global minutiae sets are generated based on the link selection procedure described in Algorithm 2.4. As shown in Figure 2.9, the reconstruction accuracy steadily improves as the number of candidates generated is increased. With a security threshold of 2^{50}, i.e. when the adversary is able to try 2^{50} candidate reconstructions, the genuine accept rate for the different impression scenario is 45% at an FAR of 0.01%, which is 3% more than the genuine accept rate achieved when top-100 links are executed to reconstruct the fingerprint.

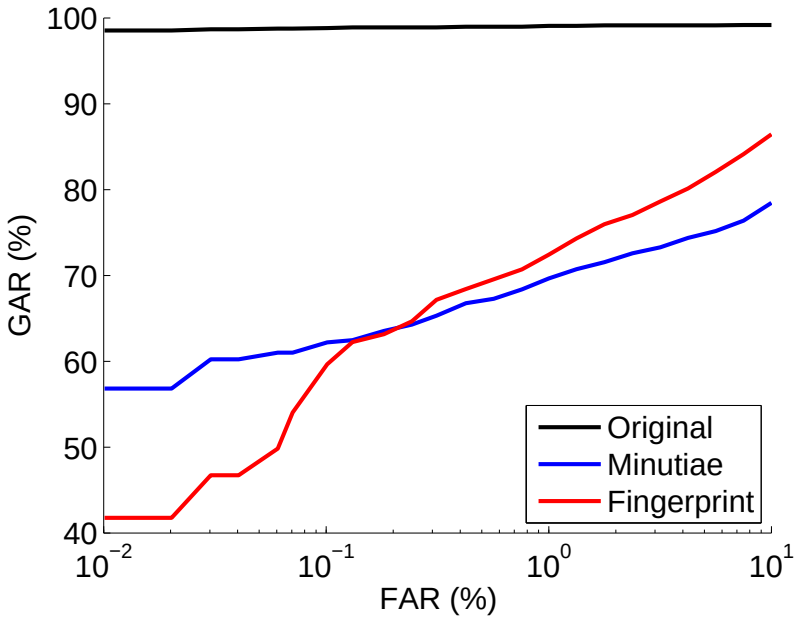


Figure 2.11: ROC curves corresponding to the cases when original fingerprint is matched with another impression of the same finger, fingerprint reconstructed using top-100 links is matched with another impression of the same finger, and the case when minutiae recovered using top-100 links are directly matched with the minutiae from another impression of the same finger.

Since the reconstruction of a fingerprint image from minutiae also introduces some noise (e.g. the presence of spurious minutiae in the reconstructed image), we tested the recognition performance when only minutiae templates were matched instead of fingerprint images. We observe that, the recognition performance obtained by the recovered minutiae was significantly better than the case when the reconstructed fingerprint images were matched. For the different impression scenario, a genuine accept rate of 57% was achieved when only minutiae were used for matching as compared to a rate of 42% when the reconstructed fingerprint images were used for matching for the case when top-100 links were used to reconstruct the fingerprint. The corresponding ROC curves are shown in Figure 2.11. Thus, given better algorithms for reconstructing fingerprint image from minutiae, and perhaps other non-minutiae information, the matching performance of the reconstructed images based on the proposed technique can be significantly improved. Regardless, an adversary can match

the recovered minutiae directly with the minutiae templates stolen from databases in order to link the users among multiple systems.

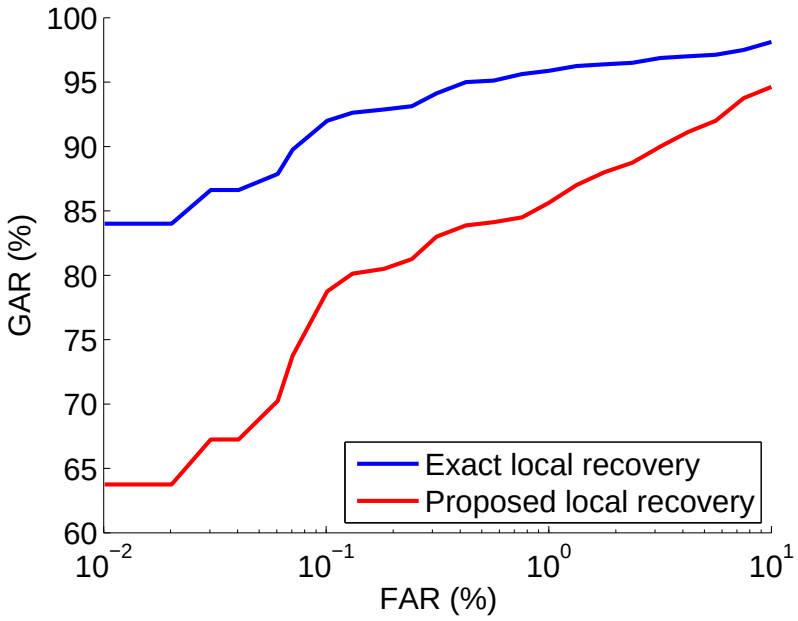


Figure 2.12: ROC curves corresponding to the cases when fingerprint reconstructed from MCC-B descriptors using top-100 links is matched with the same fingerprint and the case when the original minutiae present in the local region associated with the descriptors are used to generate the global minutiae and thus the reconstructed fingerprint.

In order to evaluate the generality of the proposed approach, we studied the hypothetical scenario where all the minutiae associated with the local MCC-B descriptors are accurately reconstructed and there are no spurious minutiae. Note that this scenario corresponds to the case when the minutiae are explicitly stored as descriptors. In this situation, the genuine accept rate of 84% is achieved compared to 64% when the proposed algorithm is used to reconstruct local MCC-B descriptors at an FAR of 0.01%. These results are for the same impression scenario when top-100 links are executed to reconstruct the fingerprint image. This difference of 20% in GAR can be attributed to i) limited information available in the MCC-B descriptors and ii) limitation of the proposed local reconstruction approach. Note that the difficulty in global minutiae recovery is due to the fact that there are many ways in which the locally

recovered minutiae set can be connected to form the global structure. In other words, there exist a large number of invalid links, and searching for the correct configuration is a difficult task. Further, there may be cases where the local neighborhood of a minutia descriptor does not overlap with any other descriptor's neighborhood. Such descriptors are not accommodated in the proposed global minutiae recovery and they are currently ignored.

2.6 Summary

The security of stored biometric templates is a critical issue in ensuring the integrity of a biometric system and to preserve user privacy. It is essential to ensure that if an impostor is able to access a biometric template, it will be extremely difficult for him to create biometric spoofs and compromise the system. It has already been shown in the literature that it is indeed possible to reconstruct a fingerprint image from its stored minutiae template thereby voiding any claims about the security of the minutiae templates. Here, we determine and analyze whether it is possible to reconstruct the minutiae and hence the fingerprint image from a minutia descriptor-only representation. Note that minutia descriptors capture information in a local neighborhood of a minutia and they do not explicitly store minutiae information (position and orientation). We show that it is indeed possible to reliably recover sufficient information about the minutiae from a descriptor-only representation of the minutiae and we quantify the success rate of an adversary attempting to reconstruct the fingerprints from the stolen templates and compromise the biometric systems. See Table 2.3. This result suggests that the biometric system designers are well advised to store the fingerprint templates in a secure manner.

In the subsequent chapters of this dissertation, we shall develop techniques to secure the biometric templates so that little biometric information can be gleaned

Scenario	Same Impression		Different Impression	
	FAR=0.01%	FAR=0.1%	FAR=0.01%	FAR=0.1%
Largest Local Minutiae Set	25%	43.5%	16%	30%
Top-100 Links	64%	79%	42%	59.5%
2^{50} Trials	67%	83%	45%	64%
Selected Templates (Top-100 Links)	73.5%	87%	52%	68.5%
Accurate Local Reconstruction (Top-100 Links)	84%	92%	65%	78.5%

Table 2.3: GAR values at FAR values of 0.1% and 0.01% for the four different scenarios considered in this chapter.

from the protected template while at the same the system should be able to recognize a query biometric presented by a genuine user. Specifically, the Chapters 3, 4, and 5 are devoted to developing biometric template protection techniques based on biometric cryptosystem whereas Chapter 6 analyzes the various template transformation techniques for protecting biometric templates.

Chapter 3

Biometric Cryptosystems

3.1 Introduction

From our analysis presented in the previous chapter, it is clear that the biometric templates in the form currently used in practice are highly vulnerable. An adversary can possibly recover the biometric image given a template which can pose serious threats to the system security as well as user's privacy. This highlights the need to develop robust template protection techniques.

In this chapter we analyze biometric cryptosystems that are one of the two main categories of software based template protection techniques. The other being the set of template transformation techniques. During the enrolment stage of a typical biometric cryptosystem, the biometric template is essentially encoded with a key to generate the protected template, also called the secure sketch, which reveals no significant information about either the associated biometric template or the key. During authentication, the biometric query is used to decode the secure sketch and thus recover the key as well as the enrolled biometric template. Note that the associated key is not explicitly stored anywhere in the system and the verification of the recovered key or the recovered enrolled biometric template is performed by encrypting them

and comparing with their encrypted form stored during enrolment. In addition to concealing the biometric template, a biometric cryptosystem also serves as a mechanism to secure a key that may be used in another cryptosystem. Note that the requirement to keep the encryption/decryption key secure is one of the major issues plaguing the current cryptographic security systems. In this chapter, we mainly focus on two commonly used biometric cryptosystems: fuzzy vault and fuzzy commitment. A fuzzy vault is designed to secure biometric templates represented as an unordered set of points whereas a fuzzy commitment is designed to secure biometric templates represented as binary vectors.

Section 3.2 describes the common biometric cryptosystems available in the literature. Section 3.3.1 describes the proposed implementation of fuzzy vault whereas implementation of fuzzy commitment is detailed in Section 3.3.2. Section 3.4 provides the security analysis of biometric cryptosystems. The three biometric traits used for experiments and their corresponding feature extraction procedure is detailed in Section 3.5. Finally, Section 3.6 provides the summary of the discussion.

3.2 Background

A simple example of a biometric cryptosystem is based on quantization of biometric features. Given an enrolled template, $\mathbf{x}^E$, that is represented as a real vector, the protected template $f(\mathbf{x}^E; q) = (\mathbf{x}^E - \mathbf{x}_q^E)$ consists of the difference between $\mathbf{x}^E$ and its quantized version $\mathbf{x}_q^E$. Each element of the quantized vector $\mathbf{x}_q^E$ is obtained as

$$\mathbf{x}_q^E(i) = arg\min_t |\mathbf{x}^E(i) - t|, t \in \{kq | k \in \mathbb{Z}\}. \tag{3.1}$$

where q denotes the width of each quantum associated with each element of $\mathbf{x}^E$ and $\mathbb{Z}$ is the set of integers. Note that $f(\mathbf{x}^E; q)$ reveals little information about the original template $\mathbf{x}^E$ if q is sufficiently small.

The authentication requires recovery of $\mathbf{x}^E$ given $f(\mathbf{x}^E; q)$ and the query biometric $\mathbf{x}^A$. Since $\mathbf{x}^A$ is expected to be similar, but not exactly same, to $\mathbf{x}^E$, the quantized version of the template, i.e. $\mathbf{x}_q^E$, is first easily constructed. For this, we subtract the protected template $f(\mathbf{x}^E; q)$ from the query $\mathbf{x}^A$ to move it close to the $\mathbf{x}_q^E$. The shifted query $\mathbf{x}_-^A = \mathbf{x}^A - f(\mathbf{x}^E; q)$ is then quantized to obtain $\mathbf{x}_q^A$. If each element of $\mathbf{x}^A$ is within a distance of $q/2$ from each corresponding element of $\mathbf{x}^E$, then $\mathbf{x}_q^A = \mathbf{x}_q^E$. This can be verified by generating a cryptographic hash of the vector $\mathbf{x}_*^A = \mathbf{x}_q^A + f(\mathbf{x}^E; q)$ and matching it with the stored hash value of $\mathbf{x}^E$. See [131] for a similar implementation of biometric cryptosystem. Note that in this cryptosystem there is no key associated with the biometric and the authentication is performed by recovering the original biometric template exactly.

An external key κ_c, represented as a binary string of the same length as that of $\mathbf{x}^E$, can also be used in this cryptosystem to modulate the different dimensions or elements of the quantum center vector i.e. $\mathbf{x}_q^E$. Here, the set of possible values for the elements of the quantum center is divided into two sets of alternate points $S_0 \equiv \{2kq | k \in \mathbb{Z}\}$ and $S_1 \equiv \{(2k+1)q | k \in \mathbb{Z}\}$. During enrolment, the ith element of $\mathbf{x}^E$ i.e. $\mathbf{x}^E(i)$ is shifted to the closest point in $S_{K(i)}$ to obtain the quantized vector $\mathbf{x}_q^E$. The protected template is obtained in the same manner as before i.e. $f(\mathbf{x}^E, \kappa_c; q) = \mathbf{x}^E - \mathbf{x}_q^E$. During authentication, the shifted query is similarly obtained as $\mathbf{x}_-^A = \mathbf{x}^A - f(\mathbf{x}^E, \kappa_c; q)$ and each element of $\mathbf{x}_-^A$ is quantized to the nearest multiple of q to obtain $\mathbf{x}_q^A$. The ith element of the key is recovered as 0 if $\mathbf{x}_q^A(i) \in S_0$ and 1, otherwise. The enrolled biometric template is recovered as $\mathbf{x}_*^E = \mathbf{x}_q^A + f(\mathbf{x}^E, \kappa_c; q)$ which can be verified using the stored hashed enrolled template. See [77] for a detailed discussion. Note that, however simple, the above approach leads to a large error rate essentially due to the sub-optimality of the quantization procedure involved. The above procedure is also referred to as the helper data extraction technique since the stored data i.e. $f(x^E, \kappa_c; q)$ helps in extraction of the key given the query biometric.

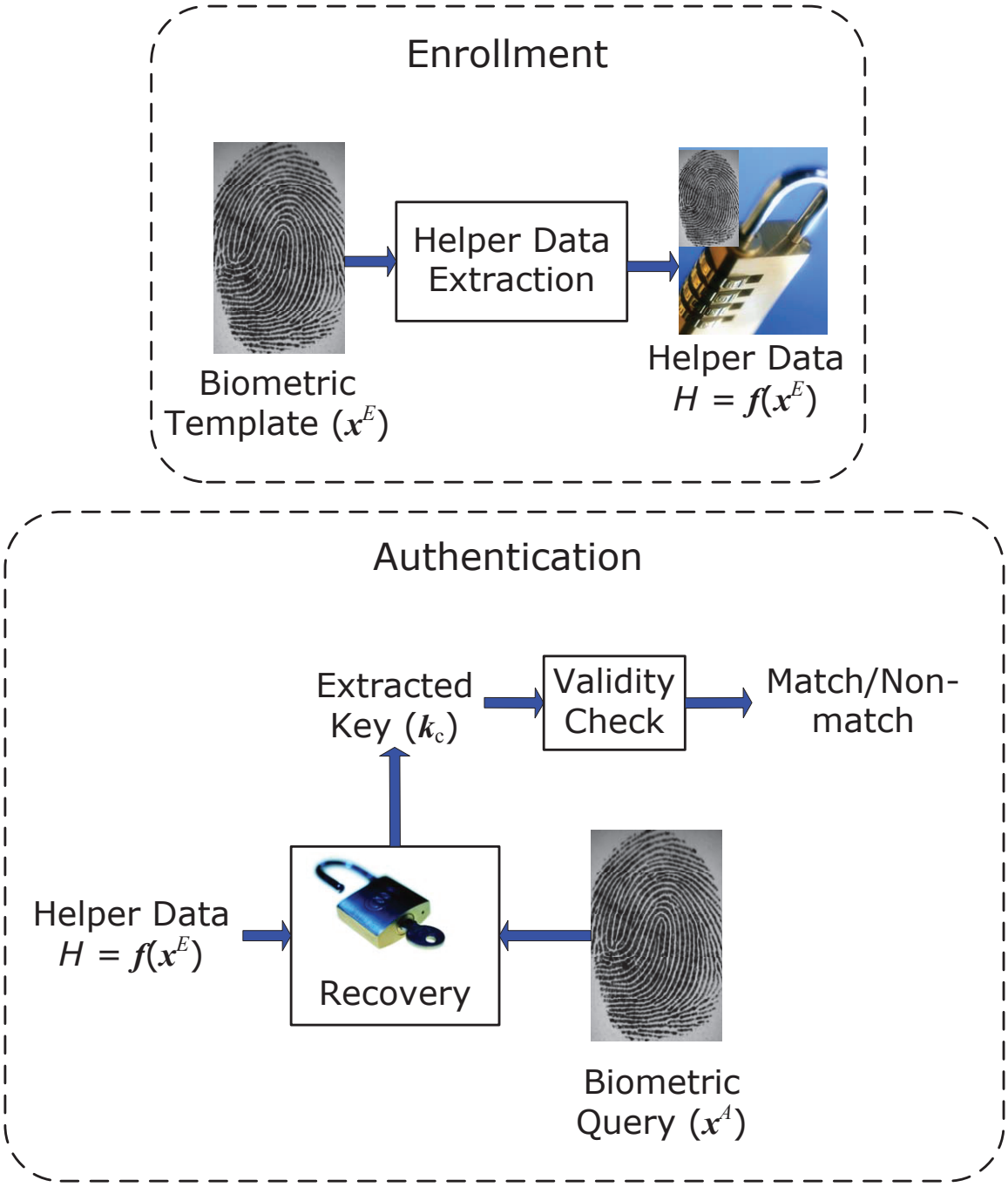


Figure 3.1: A schematic diagram for a typical biometric cryptosystem. The schematic diagram of a biometric cryptosystem is also shown in Figure 1.7.

The stored data $f(x^E, \kappa_C; q)$ is also sometimes referred to as the helper data or a secure sketch. Figure 3.1 shows a schematic diagram of a typical biometric cryptosystem.

Fuzzy commitment [69] and fuzzy vault [68] are two of the more practical biometric cryptosystems. Fuzzy commitment [69] is a biometric cryptosystem that can be used to secure biometric traits represented in the form of binary vectors (e.g. iriscodes). In the enrolment stage of a typical fuzzy commitment, a key present in the form of a codeword from a binary error correcting code is element-wise xored with the binary biometric template to obtain the secure sketch. While, during authentication, the binary query biometric is XOR'ed with the secure sketch to obtain a corrupted version of the codeword which is then decoded to recover the key. Figure 3.2 provides a schematic diagram of a typical fuzzy commitment scheme.

Fuzzy vault [68] is useful for securing point-set based biometric features such as fingerprint minutiae. In the enrolment stage of a typical fuzzy vault, each point associated with the biometric query is embedded in a finite field and is evaluated on a polynomial in the same finite field that is indexed by the key. The biometric points and their polynomial are secured by adding a large number of randomly generated points. During authentication, since the query would have similar points compared to the template, the true biometric points in the vault can are identified and are used to reconstruct the polynomial. See Figure 3.3 for an illustration of the encoding and decoding procedures of a typical fuzzy vault. Table 3.1 summarizes the comparative characteristics of fuzzy vault and fuzzy commitment.

3.3 Biometric Cryptosystem Implementation

In this section, we discuss the implementation details of fuzzy commitment and fuzzy vault techniques while highlighting the differences from certain tradition implementations. Both fuzzy vault and fuzzy commitment schemes typically use linear error

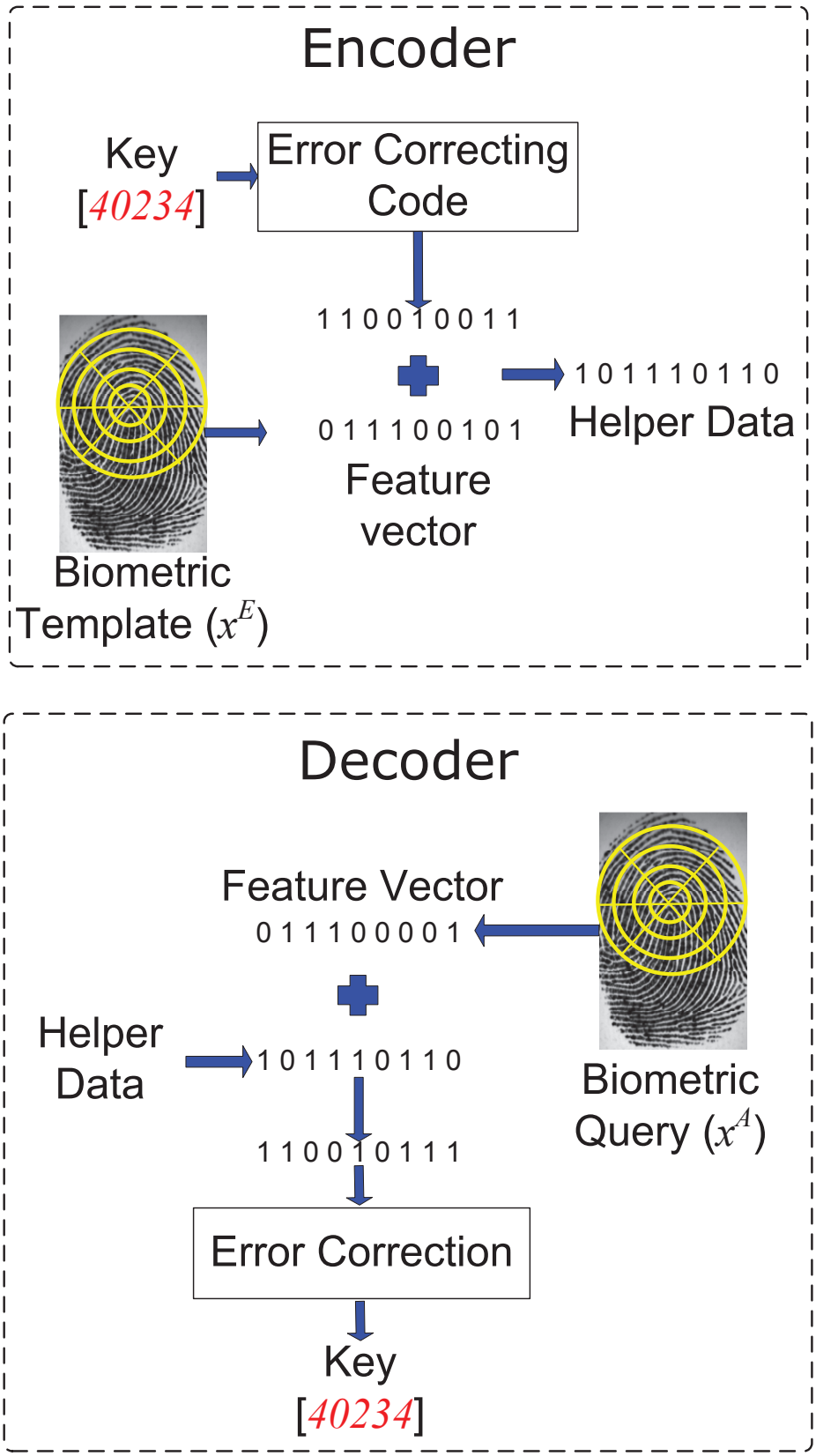


Figure 3.2: A schematic diagram illustrating encoding and decoding of a typical fuzzy commitment scheme.

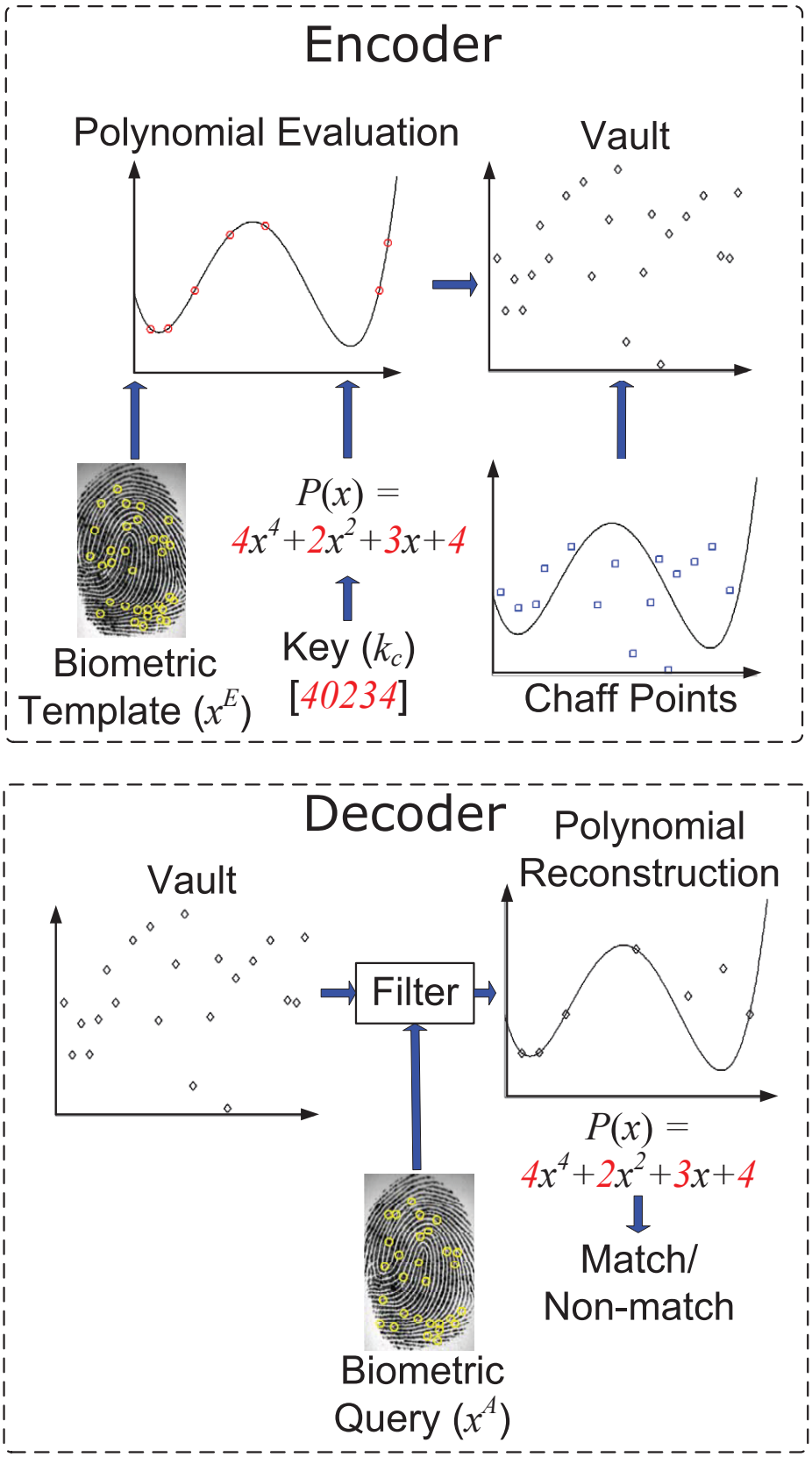


Figure 3.3: A schematic diagram illustrating encoding and decoding of a typical fingerprint fuzzy vault.

Table 3.1: Comparison of fuzzy commitment and fuzzy vault.

	Fuzzy Vault	**Fuzzy Commitment**
Representation	Point-set	Binary string
Main advantage	Ability to secure fingerprint minutiae	Compact size of the sketch
Main limitation	Difficult to generate chaff that are indistinguishable from genuine points	Lack of perfect codes for desired code lengths
Parameters	Polynomial degree (k), size of the template set (r), and number of chaff points (q)	Key length L, length of codeword N, and error correcting capacity of the code
GAR-Security tradeoff	Higher values of (k/r) and q lead to lower GAR, but higher security and vice versa	Higher values of (L/N) lead to lower GAR, but higher security and vice versa
Implementations	Fingerprint ([91, 139]), face ([44]), iris ([75]), signature ([47])	Fingerprint ([20]), face ([20, 71]), iris ([54]), signature ([80])

correcting codes. Consider a linear error correcting code of length ℓ_n (number of symbols in the codeword) and rank ℓ_k (number of symbols in the secret key). A linear error correcting code can correct any combination of g erasures and e errors as long as $(g+2e+1) \leq D_{min}$, where D_{min} is the minimum distance between the codewords of the code [53]. When such a code is employed in a biometric cryptosystem, the secure sketch can be decoded as long as $(\ell_n - D_{min} + 1)$ symbols in the biometric feature vector can be guessed correctly and the remaining $(D_{min} - 1)$ symbols are treated as erasures. If the selected error correcting code is maximum distance separable (i.e., it satisfies the Singleton bound), then $(D_{min} - 1) = (\ell_n - \ell_k)$.

As the error correction decoder in a biometric cryptosystem is generally constrained to run in polynomial-time. This approach has two limitations. Firstly, it restricts the number of errors that can be corrected to $(D_{min} - 1)/2$, thereby leading to more false rejects for genuine users. Given the large intra-user variations in biometric features, it is often difficult to find codes with sufficient error correction capability

that can provide high GAR. Secondly, the above approach requires analysis of two separate attack strategies: (i) a false accept attack, where the attacker attempts to decode a given secure sketch by invoking the polynomial-time decoder multiple times with different non-matching queries from a database, and (ii) a brute-force attack, where the attacker directly tries to guess $(\ell_n - D_{min} + 1)$ symbols in the original biometric feature vector. It is not clear which strategy is more efficient from the attacker's perspective.

In this dissertation, we modify the existing implementations of biometric cryptosystems to relax the constraint that the decoder needs to run in polynomial-time. During each iteration of our decoding algorithm, we consider only a subset of most reliable symbols from the codeword and attempt to decode the sketch by considering the remaining symbols as erasures. If the sketch cannot be decoded in a particular iteration, we attempt to decode it using a smaller subset of symbols with minimum size $(\ell_n - D_{min} + 1)$. With this decoding procedure, the sketch will be eventually decoded for every authentication query. However, the decoding complexity will be different for the genuine and impostor cases. In practice, one can set a threshold on the decoding complexity for genuine users and measure GAR as the fraction of genuine authentication attempts where the decoding complexity is less than the selected threshold. The security is measured as the minimum computational complexity faced by the attacker for a successful decoding among the various impostor match attempts. Thus, the proposed security metric takes into account both the false accept (number of impostor attempts needed) and brute-force attack (minimum complexity of an impostor attempt) strategies.

3.3.1 Fuzzy Vault Implementation

Fuzzy Vault Encoding

Let $\mathbf{s}^E = \{u_i\}_{i=1}^r$ be the biometric template represented as a set of r points, which is to be secured using a vault. Let $\mathbf{U}$ be the universe of all possible biometric points. To construct a vault, each point in $\mathbf{U}$ is assigned[1] to a point from a finite field $\mathcal{F}$. Let x_i be the element in $\mathcal{F}$ associated with the point u_i in $\mathbf{s}^E$, $\forall\ i = 1, 2, \cdots, r$ and let $\mathbf{s}_g^E = \{x_i\}_{i=1}^r$. A set of q chaff points are randomly selected from $(\mathbf{U} \setminus \mathbf{s}^E)$ (' $\setminus$ ' denotes the set difference operator). Let $\mathbf{s}^C = \left\{u_j^*\right\}_{j=1}^q$ be the set of chaff points and let $\mathbf{s}_g^C = \left\{x_j^*\right\}_{j=1}^q$ be the corresponding set of points obtained by mapping elements in $\mathbf{s}^C$ to elements in $\mathcal{F}$. Given a key κ_c of length L bits, we encode it as a polynomial P of degree k. Finally, the vault is obtained as a set of 3-tuples as follows: $\mathbf{y}_c = \{(\alpha_i, \beta_i, \gamma_i)\}_{i=1}^t$, where $t = (r + q)$, $\alpha_i \in (\mathbf{s}^E \cup \mathbf{s}^C)$, β_i is the corresponding element in $(\mathbf{s}_g^E \cup \mathbf{s}_g^C)$, and γ_i is given by

$$\gamma_i = \begin{cases} P(\beta_i), & \text{if } \alpha_i \in \mathbf{s}^E, \\ \\ b_i, \text{where } b_i \in \mathcal{F} \setminus \{P(\beta_i)\}, & \text{if } \alpha_i \in \mathbf{s}^C. \end{cases} \tag{3.2}$$

Fuzzy Vault Decoding

During authentication, the query $\mathbf{s}^A = \{u'_j\}_{j=1}^{r'}$ is used to identify the genuine points in the vault; if the query overlaps sufficiently with $\mathbf{s}^E$ then the polynomial P could be correctly reconstructed. The number of points correctly identified as genuine in the vault that are sufficient for decoding depends on the decoding procedure used. In case the simple Lagrange interpolation is used for decoding, i.e. all possible sets of points with cardinality $k + 1$ are used to reconstruct the polynomial using Lagrange interpolation, the decoding will be successful as long as more than k points have

[1] This mapping can be stored as a lookup table or defined by a hash function.

been correctly identified in the vault as potential candidates for genuine points. The complexity of this decoding will, however, depend on the number of chaff points identified as genuine. In case a Berlekamp-Massey algorithm is used for decoding, the vault will be correctly decoded if the number of correctly identified genuine points is at least $(n + k + 1)/2$, where n is the total number of points identified as candidate genuine points. Usually, a hash of the key κ_c is stored in the system and the hash of the key recovered during authentication is matched with it to verify its correctness. One limitation of the decoding procedures described above is that the candidate genuine points are identified only once per decoding and various subsets of this set are used. To overcome this limitation, we propose an improved decoding procedure which is described below.

In the improved decoding procedure, for each point α_i $(i = 1, 2, \cdots, t)$ in the vault, its distance to the closest query point is computed and the list of vault points is sorted based on this distance. The ordered set of points in the vault is given by $\mathbf{y}_c^o = [(\alpha(1), \beta(1), \gamma(1)), \cdots, (\alpha(t), \beta(t), \gamma(t))]$, where $\min_w d(\alpha(i), u'_w) < \min_w d(\alpha(j), u'_w)$ if $i < j$, and $w \in \{1, \cdots, r'\}$. Finally, the Berlekamp-Massey[2] (B-M) algorithm [14] is applied on subsets of different lengths derived from $\mathbf{y}_c^o$ to decode the vault and thereby recover the associated polynomial and the key κ_c (see Algorithm 3.1).

Algorithm 3.1 is based on the following principle. Given a set of n points from the vault, the Berlekamp-Massey decoding allows recovery of the polynomial if there are at least $(n + k + 1)/2$ genuine points in the given set. Since the points in the vault are ordered according to their likelihood of being genuine, we consider subsets of n $((k + 1) \leq n \leq t)$ most likely points in parallel. If a selected subset of length n cannot decode the vault, some points in the subset are randomly removed to obtain smaller subsets of minimum size $(k + 1)$. Since all points in the vault are used in

[2]The Berlekamp-Massey (B-M) algorithm is one of the well-known decoding algorithms used for Reed-Solomon codes.

[3]**forall** is the parallel for-loop; all instances of the loop run in parallel

Algorithm 3.1 Fuzzy vault decoding based on Berlekamp Massey algorithm [14].

Input: $\mathbf{y}_c^o = [(\alpha(1), \beta(1), \gamma(1)), \cdots, (\alpha(t), \beta(t), \gamma(t))]$ (Ordered vault points); k (Degree of polynomial)
forall $^3 n = (k+1)$ to t **do**
  $\mathbf{s}_n \leftarrow \{(\alpha(i), \beta(i), \gamma(i))\}_{i=1}^{n}$
  for $m = 0$ to $n - (k+1)$ **do**
    forall $\mathbf{s}_* \subset \mathbf{s}_n, |\mathbf{s}_*| = m$ **do**
      $\mathbf{s}_n^- \leftarrow \mathbf{s}_n \setminus \mathbf{s}_*$
      $P \leftarrow DecodeBM(\mathbf{s}_n^-, k)$
      if P is the required polynomial **then**
        Return P
      end if
    end forall
  end for
end forall
Return ϕ
{$DecodeBM(\mathbf{s}, k)$ performs a Berlekamp-Massey decoding of the set of points $\mathbf{s}$ for a polynomial of degree k}

decoding, the vault will always be eventually decoded, but the decoding complexity will be different for each query. Since the points in the vault are ordered based on their distance to the points in the query biometric set, one would expect the decoding complexity for a genuine user to be significantly less than the decoding complexity for an impostor.

3.3.2 Fuzzy Commitment Implementation

In the case of fuzzy commitment, we assume that the enrolled biometric template $\mathbf{b}^E$ is an N-bit binary string. In order to generate a fuzzy commitment, a uniformly random key κ_c of length L ($L \leq N$) bits is generated and used to uniquely index a N-bit codeword $\mathbf{c}$ of an appropriate error correcting code. The sketch is then extracted from the template as $\mathbf{y}_c = \mathbf{c} \oplus \mathbf{b}^E$, where $\oplus$ indicates the modulo-2 addition. The sketch $\mathbf{y}_c$ is stored in the database along with $\mathbf{h}(\kappa_c)$, where $\mathbf{h}(.)$ is a cryptographic hash function. During authentication, the codeword is obtained from the query biometric $\mathbf{b}^A$ and the sketch $\mathbf{y}_c$ as follows: $\mathbf{c}^* = \mathbf{y}_c \oplus \mathbf{b}^A = \mathbf{c} \oplus (\mathbf{b}^E \oplus \mathbf{b}^A)$. This codeword

$\mathbf{c}^*$, which is generally a corrupted version of the original codeword $\mathbf{c}$, can be decoded to get the key κ^*. The authentication is deemed successful if $\mathbf{h}(\kappa^*)$ is the same as $\mathbf{h}(\kappa_C)$. If the Hamming distance between $\mathbf{b}^E$ and $\mathbf{b}^A$ is not greater than the error correcting capacity of the code, κ^* would be the same as κ and the matching will be successful.

We improve this basic procedure to decode the fuzzy commitment also in a similar manner as the fuzzy vault. Algorithm 3.2 provides the improved fuzzy commitment decoding procedure. If the error (crossover) probabilities of each bit in the biometric feature vector is known, it is possible to consider some of the least reliable bits as erasures during decoding. As in the case of fuzzy vault, we consider the n most reliable bits in parallel ($(N - D_{min} + 1) \leq n \leq N$) and treat the remaining bits as erasures. If the decoding is still not successful, a subset of reliable bits of size m are flipped. If the number of errors among the flipped bits is more than $(m/2)$, then the number of errors in the selected set of reliable bits will be less after flipping, thereby increasing the possibility of successful decoding.

3.4 Methodology for Security Analysis

While information-theoretic measures such as entropy loss or leakage rates are typically used to characterize the security of biometric cryptosystems, such measures are difficult to estimate when the precise distribution of biometric features is not known. In practice, unrealistic assumptions about the biometric features (e.g., uniform distribution) are used to estimate the leakage rates, which provide only loose upper bounds on the security [25, 87]. To account for this factor, we assume that the attacker has access to a large biometric database (analogous to a dictionary attack in password-based systems). We then empirically estimate the security based on the minimum decoding complexity among all impostor matches tried by the attacker to

Algorithm 3.2 A fuzzy commitment decoding algorithm that allows for erasures in the codeword based on the crossover probabilities.

Input: $\mathbf{c}^*$ (corrupted codeword); $\mathbf{p} = [p_1, \cdots, p_N]$ (bit reliability vector where p_i indicates the reliability (1-crossover probability) of $\mathbf{c}^*(i)$, $i = 1, 2, \cdots, N$); D_{min}.
forall $n = (N - D_{min} + 1)$ to N **do**
  $\mathbf{s}_n \leftarrow RBS(\mathbf{p}, n, N)$
  for $m = 0$ to $D_{min} + 1$ **do**
    forall $\mathbf{s}_* \subset \mathbf{s}_n, |\mathbf{s}_*| = m$ **do**
      $\mathbf{c'} \leftarrow Flip(\mathbf{c}^*, \mathbf{s}_*)$
      $\kappa_c \leftarrow DecodeFC(\mathbf{c'}, \mathbf{s}_n, L)$
      if κ_c is the required key **then**
        Return κ_c
      end if
    end forall
  end for
end forall
Return ϕ
{$DecodeFC(\mathbf{c'}, \mathbf{s}_n, L)$ is an error correction decoder that corrects the errors in the corrupted codeword $\mathbf{c'}$ to obtain a key of length L, while considering all bits whose indices are not indicated in $\mathbf{s}_n$ as erasures. The function $RBS(\mathbf{p}, n, N)$ returns the indices of the n most reliable bits. $Flip(\mathbf{c}^*, \mathbf{s}_*)$ returns the codeword $\mathbf{c'}$, in which the bits in $\mathbf{c}^*$ corresponding to points in $\mathbf{s}_*$ are flipped.}

decode a given secure sketch. While estimating the computational complexity, we assume that the complexity of the error correction decoder (e.g., B-M algorithm) is unity, and consider only the number of times this decoder needs to be invoked. The proposed security measure is a "product" of the number of impostor matching attempts (related to false accept attacks) and the minimum decoding complexity of an impostor matching attempt (related to brute force attacks). Thus, we combine the two attack strategies traditionally used to estimate system security. Furthermore, during authentication, the symbols in the codeword are ordered based on the query prior to decoding. Therefore, the proposed security measure indirectly takes into account the distribution of biometric features and provides a more reliable estimate of the difficulty in breaking a secure sketch, which is usually greater than $-log$(FAR) bits.

3.4.1 Fuzzy Vault Security

Traditionally, the security of a fuzzy vault is measured based on the assumption that an attacker will conduct a brute force attack on the vault by selecting a set of $k+1$ points from the vault and use them to reconstruct the polynomial of degree k using Lagrange interpolation procedure. The security against such an attack is measured as

$$\mathcal{S}_{FV} = \frac{\binom{t}{(k+1)}}{\binom{r}{(k+1)}} \tag{3.3}$$

where t is the total number of points in the vault and r is the number of genuine points in the vault. The security analysis of the improved decoding procedure proposed in Section 3.3.2 is described below.

Suppose that the attacker has access to $\mathcal{N}_I$ impostor samples to decode a vault ($\mathbf{y}_C$). Let $\mathbf{s}_n^I$ denote a set containing the first n points from the ordered set of vault points ($\mathbf{y}_C^o$). Here, the ordering is based on the distance of the vault points to the points in the query biometric set from impostor I. Let r_n^I be the number of genuine points in $\mathbf{s}_n^I$, i.e., $r_n^I = |\mathbf{s}_n^I \cap \mathbf{s}^E|$, where $\mathbf{s}^E$ is the enrolled template secured using $\mathbf{y}_C$. For $(k+1) \leq n \leq t$, where t is the total number of points in the vault, three different scenarios are possible.

1. If $r_n^I \geq (n+k+1)/2$, the B-M algorithm will return the correct polynomial in a single attempt.

2. If $(k+1) \leq r_n^I < (n+k+1)/2$, one needs to find the minimum value of m_n^I such that when m_n^I chaff points are removed from $\mathbf{s}_n^I$, r_n^I becomes greater than $((n-m_n^I)+k+1)/2$. Hence, $m_n^I = \max(0,(n-2r_n^I+k+1))$ and the corresponding complexity is approximately $\binom{n}{m_n^I}/\binom{n-r_n^I}{m_n^I}$.

3. If $r_n^I < (k+1)$, the vault cannot be decoded using $\mathbf{s}_n^I$. In this case, the corresponding value of complexity is considered to be ∞.

Based on the above analysis, the security of the vault can be expressed as

$$\begin{aligned} \mathcal{S}_{FV} &= \min_{n,I} \left(\log_2 \sum_{i=0}^{m_n^I} \frac{\binom{n}{i}}{\binom{n-r_n^I}{i}} \right) + \Omega \\ &\approx \min_{n,I} \left(\log_2 \frac{\binom{n}{m_n^I}}{\binom{n-r_n^I}{m_n^I}} \right) + \Omega, \end{aligned} \tag{3.4}$$

where $\Omega = \log_2 (\mathcal{N}_I(t-k))$. The first term in Eq. (3.4) measures the complexity of a brute-force attack by an impostor and is minimized over all impostor samples. Therefore, adding more impostors is likely to lower this term. However, adding more impostors (false accept attack) will also increase the number of computations needed, which is reflected by the Ω term. An increase in the polynomial degree k will increase n and consequently result in higher security.

Since the decoding algorithm is common to both the genuine user and the impostor, we can also estimate the decoding complexity for a genuine match. Let $\mathbf{s}_n$ denote a set containing the first n points from the ordered set of vault points ($\mathbf{y}_C^o$), where the ordering is based on the distance of the vault points to the points in the query from the genuine user. Let r_n be the number of genuine points in $\mathbf{s}_n$, i.e., $r_n = |\mathbf{s}_n \cap \mathbf{s}^E|$. The decoding complexity for the genuine user can be expressed as

$$\mathcal{S}_{FV}^{gen} \approx \min_{n} \left(\log_2 \frac{\binom{n}{m_n}}{\binom{n-r_n}{m_n}} \right) + \log_2 (t-k), \tag{3.5}$$

where $m_n = \max(0, (n - 2r_n + k + 1))$.

3.4.2 Fuzzy Commitment Security

Traditionally, the security of a fuzzy commitment is measured based on the rank of the error correcting code used in constructing the fuzzy commitment. However, this technique does not consider the non-uniform distribution of biometric features. To decode a fuzzy commitment sketch, one needs to guess the bits in the binary template $\mathbf{b}^E$. Though the length of the template $\mathbf{b}^E$ is N bits, the entropy[4] of the template (N_*) is typically much less than N bits. This is because some bits may not be uniformly distributed (0 and 1 values are not equally likely), while there may also be correlation between the bits. The proposed analysis of security of a fuzzy commitment is described below.

Suppose that the attacker has access to $\mathcal{N}_I$ impostor samples and a sketch $\mathbf{y}_c$. For each impostor I, a corrupted codeword $\mathbf{c}^I$ is obtained as $(\mathbf{y}_c \oplus \mathbf{b}^I)$, where $\mathbf{b}^I$ is the binary feature vector from impostor I. Let $\mathbf{s}_n$ denote a set containing the indices of the n most reliable bits in the biometric template[5]. Let $\mathbf{b}^E_n$, $\mathbf{b}^I_n$, and $\mathbf{c}^I_n$ be substrings of $\mathbf{b}^E$, $\mathbf{b}^I$, and $\mathbf{c}^I$, respectively, containing only those bits whose indices are in $\mathbf{s}_n$. The Hamming distance between $\mathbf{b}^E_n$ and $\mathbf{b}^I_n$ is denoted as ρ^I_n.

Let $DecodeFC(\mathbf{c}^I, \mathbf{s}_n, L)$ be the error correction decoder that corrects the errors in the corrupted codeword $\mathbf{c}^I$ to obtain a key of length L while considering all bits whose indices are not in $\mathbf{s}_n$ as erasures. When the attacker invokes the above error correction decoder for values of n in the range $[N - D_{min} + 1, N]$, where D_{min} is the minimum distance of the code, three different scenarios are possible.

1. The values of n and ρ^I_n are such that $((N - n) + 2\rho^I_n) \leq (D_{min} - 1)$, where $(N - n)$ is the number of erasures and ρ^I_n is the number of errors. In this case, the decoder will return the correct key in a single attempt.

[4] We use a procedure similar to the one used in [35] to estimate the entropy. See Appendix A for details.

[5] We assume that the attacker can somehow estimate the bit reliability vector (i.e., the crossover probability for each bit in the biometric template).

2. If $((N-n)+2\rho_n^I) > (D_{min}-1)$, the attacker can try to find m_n^I ($0 \leq m_n^I \leq ((D_{min}-1)-(N-n))/2 = (n-L)/2$) such that, when m_n^I errors are corrected from $\mathbf{c}_n^I$, $((N-n)+2(\rho_n^I - m_n^I))$ becomes less than or equal to $(D_{min}-1)$. If such an m_n^I exists, then its minimum value is given by $m_n^I = \max(0, (((N-n)-(D_{min}-1))/2+\rho_n^I))$ and the corresponding complexity is approximately $\binom{n}{m_n^I}/\binom{\rho_n^I}{m_n^I}$.

3. If no such m_n^I can be found, the secure sketch cannot be decoded by considering the least reliable $(N-n)$ bits as erasures. Hence, the corresponding value of complexity is considered to be ∞.

Based on the above analysis, the security of the fuzzy commitment scheme can be expressed as

$$\begin{aligned} \mathcal{S}_{FC} &= \min_{n,I} \left(\log_2 \sum_{i=0}^{m_n^I} \frac{\binom{n}{i}}{\binom{\rho_n^I}{i}} \right) + \Omega \\ &\approx \min_{n,I} \left(\log_2 \frac{\binom{n}{m_n^I}}{\binom{\rho_n^I}{m_n^I}} \right) + \Omega, \end{aligned} \tag{3.6}$$

where $\Omega = \log_2(\mathcal{N}_I D_{min})$. The above expression, however, assumes that the bits in $\mathbf{b}_n^E$ are independent and uniformly random. Suppose that the entropy of $\mathbf{b}_n^E$ is only n_* bits. In this case, the effective Hamming distance between $\mathbf{b}_n^E$ and $\mathbf{b}_n^I$ is $\rho_{n_*}^I = (n_*\rho_n^I)/n$ and the corresponding value of m_n^I is $m_{n_*}^I = \max(0, (((N-n)-(D_{min}-1))/2+\rho_n^I)n_*/n)$. Thus, the security is given by

$$\mathcal{S}_{FC} \approx \min_{n,I} \left(\log_2 \frac{\binom{n_*}{m_{n_*}^I}}{\binom{\rho_{n_*}^I}{m_{n_*}^I}} \right) + \Omega. \tag{3.7}$$

Suppose $\mathbf{b}^A$ is a genuine authentication query and ρ_{n*} is the effective Hamming distance between $\mathbf{b}_n^E$ and $\mathbf{b}_n^A$, where $\mathbf{b}_n^E$ and $\mathbf{b}_n^A$ are the substrings of $\mathbf{b}^E$ and $\mathbf{b}^A$, respectively, containing only the n most reliable bits. The decoding complexity for a genuine match can be expressed as

$$\mathcal{S}_{FC}^{gen} \approx \min_n \left(\log_2 \frac{\binom{n_*}{m_{n_*}}}{\binom{\rho_{n_*}}{m_{n_*}}} \right) + \log_2(D_{min}), \tag{3.8}$$

where $m_{n_*} = \max(0, (((N-n)-(D_{min}-1))/2 + \rho_n)n_*/n)$.

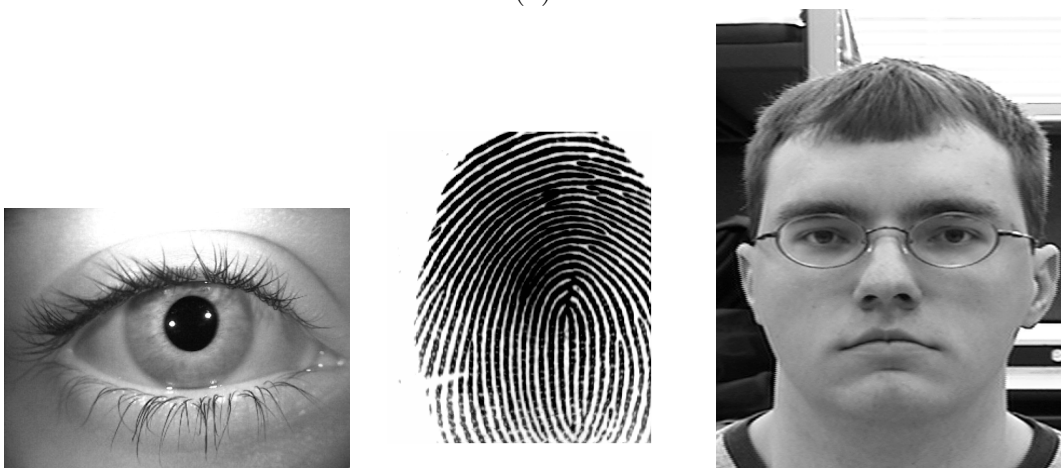
3.5 Experimental results

Here, we detail the features extracted from the finger, face and iris biometrics and quantitatively analyze their matching accuracy and security after applying biometric cryptosystems.

3.5.1 Databases

The four biometric databases used in our experiments are: the Fingerprint Verification Competition (FVC) 2002 Database-2 [79], the CASIA Iris database Ver-1 [78], the XM2VTS [5] face database, and the West Virginia University (WVU) multimodal database [32] containing fingerprints, iris and face images. See Figure 3.4 for sample images from each of these four databases. We randomly select 100 subjects each from the FVC, CASIA and the XM2VTS databases whereas 138 subjects are used from the WVU database. In our experiments, we consider one genuine authentication attempt per user and impostor attempts are simulated by using one impression of each user's biometric to authenticate as every other user. Consequently, the number of impostor attempts N_I is $9,900$ (100×99) for the virtual multimodal database and $18,906$ (138×137) for the real multimodal database.

(a)



(b)

Figure 3.4: Sample iris, fingerprint, and face images from (a) CASIA Ver-1, FVC2002 DB-2, and XM2VTS databases, respectively, and (b) WVU multimodal database. Note that the quality of iris images in the WVU database is much lower than that in the CASIA database.

Fingerprint Processing

Here, we follow [62] in processing fingerprints for constructing fuzzy vault. Since minutiae are an unordered set of points, a fingerprint is secured using a fuzzy vault. In order to construct the fingerprint fuzzy vault, a set of at most 24 good quality and well separated minutiae is selected from the given fingerprint image as the biometric points. The chaff points are randomly generated as in [91] to obtain a vault with 224 points ($r = 24, q = 200$, and $t = 224$). In addition to genuine minutiae and chaff points, points in the fingerprint corresponding to high ridge curvature are also separately stored in the system. These points were used to align the query fingerprint with the enrolled fingerprint [91]. During authentication, the query minutiae set is first aligned with the vault points using the high curvature points. A bounding box is then used to filter out points in the vault that are not in close proximity [91] of the query minutiae. The query is then further aligned with the remaining vault points using a minutiae matcher. These aligned points are then used to compute the closest distances of the vault points to the query point based on which the vault points are ordered prior to decoding using the procedure described in Section 3.3.1.

Iris Processing

Iris features are extracted in the form of a binary vector called IrisCode using the algorithm described in [115]. In case of CASIA Ver-1 database, 48 different radii and 360 different angles are used to tessellate the iris region and two bits were extracted from each region based on the response obtained from a Gabor filter. In case of WVU Iris database 20 different radii and 240 different angles are used. The complete IrisCode is thus $34,560$ and $9,600$-bits long for the CASIA Ver-1 and WVU Iris databases, respectively.

In order to reduce the dimensionality of the iriscode and remove the redundancy present in the code, Linear Discriminant Analysis (LDA) [36] is applied to the iriscode

features. Only the top 80 LDA coefficients are retained ($\ell = 80$) and these real-valued features are then binarized. For this, we quantize each element of the real-valued vector into $(\tau + 1)$ fixed size quanta, $\tau = 40$. The quantized values are then represented using τ-bit unary[6] representation in order to obtain a binary string of length $\tau\ell$, where ℓ is the dimensionality of the original vector. In the second stage, we select 1023 most discriminable bits (N). The discriminability of each bit is computed as $((1-p_g^e)p_i^e)$, where p_g^e and p_i^e are the genuine and impostor bit-error probabilities, respectively. One iris image each is used for enrolment and authentication, while the remaining samples are used as the training set in order to compute the LDA features. Due to their binary nature, the iris features are secured using fuzzy commitment scheme.

Face Processing

Alignment of face images is essential prior to feature extraction. For the WVU database, eye locations were automatically extracted using the Identix FaceIT software [3], a region of size 120×100 was cropped such that the inter-pupil distance is 60 pixels. In case of XM2VTS database, we use the FaceVACS software from Cognitec [4] to extract the eye coordinates for aligning the face images. The inter-pupil distance is set to 37.5 pixels. We then crop the aligned face image to a region of size 120×100 pixels. Histogram equalization is used to reduce the effect of illumination variations. We then extract 80 LDA coefficients ($\ell = 80$) that constitute the real-valued feature vector representing a face image. The same procedure applied to the iris LDA coefficients is also applied to the face LDA coefficients to generate a binary string and a set of $1,023$ bits are similarly extracted. Again, one face image each is

[6] A unary encoding works as follows. Suppose that a real-value a needs to be encoded using τ bits. The range of a, say $[a_{min}, a_{max}]$, is quantized into $(\tau+1)$ bins. If a falls into the i^{th} bin, it is represented as $(\tau - i + 1)$ ones followed by $(i-1)$ zeros, where $i = 1, 2, \cdots, (\tau + 1)$.

used for enrolment and authentication, while the remaining samples are used as the training set in order to compute the LDA features. Similar to iris, face features are also secured using fuzzy commitment technique.

3.5.2 Performance Evaluation

We evaluate the trade-off between recognition accuracy and security of the biometric cryptosystems using a plot between the genuine accept rate (GAR) of the system and the amount of security imparted by the system (which also considers the false accepts), called the GAR-Security (G-S) curve. The GAR is measured as the fraction of genuine authentication attempts, where the decoding complexity ($\mathcal{S}_{FV}^{gen}$ and $\mathcal{S}_{FC}^{gen}$ for fuzzy vault and fuzzy commitment, respectively) is less than 15 bits. The security is measured as the minimum computational complexity faced by the attacker for a successful decoding among the various impostor match attempts. The G-S curve is obtained by varying the length (L) of the key (κ_c) used in the biometric cryptosystem. Note that based on our formulation, the minimum value of security corresponds to the value of Ω as defined in Eq. (3.4) and Eq. (3.7) for fuzzy vault and fuzzy commitment, respectively. The value of Ω is determined by the number of impostor trials the adversary is conducting in addition to the characteristics of the decoding procedure used. Thus the adversary can modify the number of impostor trials conducted based on the availability of a database of biometric templates as well as in order to optimize the expected computation required in decoding a protected template. In case of the fuzzy commitment, the G-S curves are obtained by varying D_{min} of the error correcting code from 0.02 to 0.6 times the length of the binary string N.

Figure 3.5 shows the G-S curves corresponding to the fingerprint fuzzy vault for the FVC2002 Database-2 and the WVU databases. Note that there is around 30% difference in the GAR between the two databases at the lowest security setting. This large difference in the GAR highlights the importance of the biometric image quality,

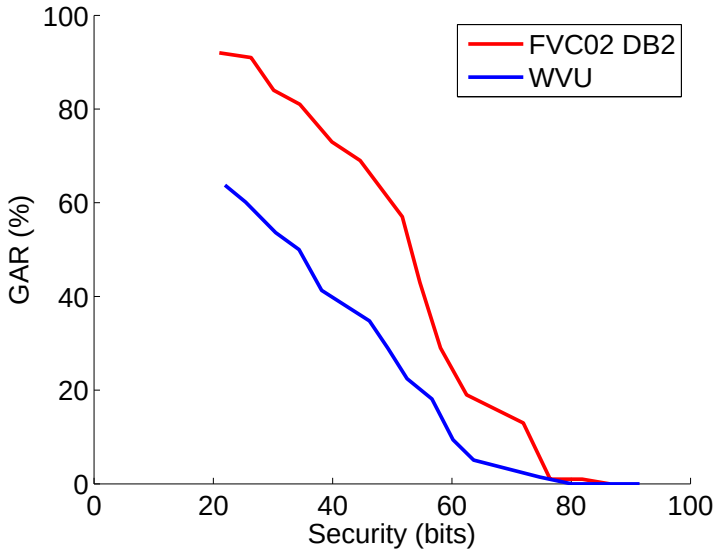


Figure 3.5: The G-S curves for fuzzy vault for fingerprints from FVC 2002 DB-2 and WVU databases.

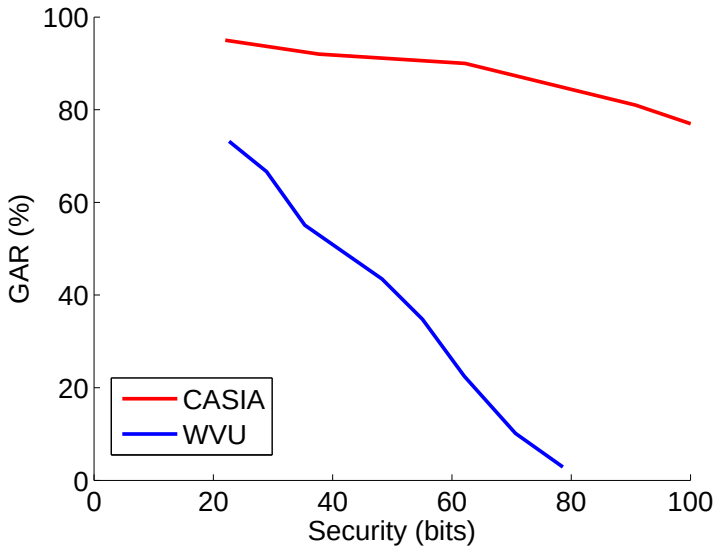


Figure 3.6: The G-S curves for fuzzy commitment for iris images from CASIA Ver-1 and WVU databases.

which in turn depends on the way in which the data is captured, in regards to the amount of the identifying information present in the captured biometric data and the amount of security it can impart. The difference in the performance among the two databases considered is largest for the iris modality. Figure 3.6 shows the G-S curves corresponding to the iris fuzzy commitment for CASIA Ver-1 and WVU databases. This is expected given significantly poor quality of iris images in the WVU database compared to those in the CASIA Ver-1 database. Since the capture of face image is relatively straightforward compared to capturing fingerprint and iris. Therefore, the difference in the security imparted by the biometric templates obtained from the two face databases is not very large. Figure 3.7 shows the G-S curves corresponding to face fuzzy commitment for XM2VTS and WVU databases. Table 3.2 compares the genuine accept rates of the different biometric cryptosystems at a security level of 53 bits, which is equivalent to the guessing entropy of a 8-character password randomly chosen from a 94-character alphabet [22]. Note that the security level of 53 bits is higher when compared to those typically reported in the literature [54, 91]. Furthermore, the proposed security measure takes into account the distribution of biometric features and hence, provides a tighter bound on the security of the sketch. Also note that these values for GAR are significantly lower compared to state of the art matching performance reported in literature. For example, the best GAR reported in case of fingerprints from FVC 2002 DB2 is 99.7% when there was no false accept [79].

3.6 Summary

In this chapter we have detailed procedures to evaluate two of the most common biometric cryptosystems namely, fuzzy vault and fuzzy commitment. We have developed a new way to evaluate the performance of a biometric cryptosystem using

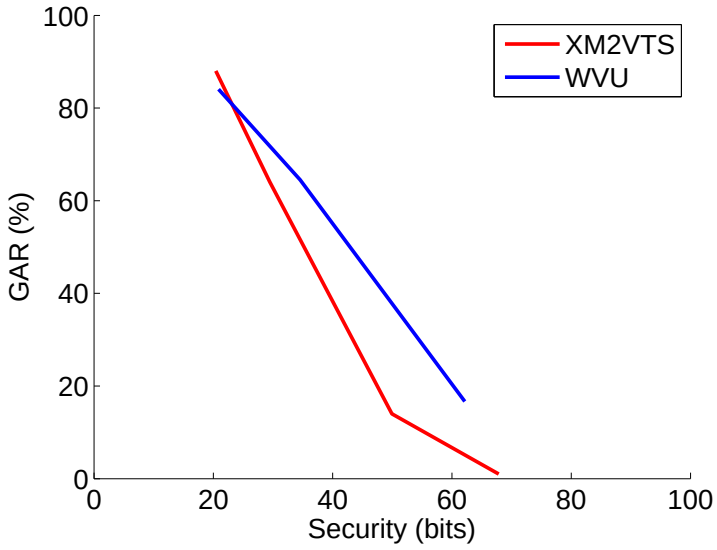


Figure 3.7: The G-S curves for fuzzy commitment for face images from XM2VTS and WVU databases.

Trait	GAR	Trait	GAR	Trait	GAR
Finger (FVC)	51%	Face (XM2VTS)	12%	Iris (CASIA)	91%
Finger (WVU)	22%	Face (WVU)	33%	Iris (WVU)	37%

Table 3.2: Comparison of genuine accept rates of the different biometric cryptosystems at a security level of 53 bits, which equals the security imparted by a randomly chosen 8 character password [22]. Note that these values for GAR are significantly lower compared to state of the art matching performance obtained reported in literature. For example, the best GAR reported in case of fingerprints from FVC 2002 DB2 is 99.7% when there was no false accept [79].

the G-S curve that depicts the trade-off between the convenience the system provides in terms of the genuine accept rate and the amount of security it ensures in terms of the complexity of the attacks an adversary can stage on the system. From our experimental results it can be seen that the use of biometric cryptosystems significantly reduces the discriminative capabilities of the biometric traits and thus new research directions should be pursued in order to further improve the accuracy of the biometric cryptosystems. Also, note that biometric cryptosystems do not provide non-linkability which is one of the two main requirements of a biometric template protection technique. See Section 1.3. In fact, the need for non-linkability was the main reason for development of template transformation techniques as discussed in Chapter 6.

One way to improve the security and matching accuracy provided by a biometric cryptosystem is to combine multiple biometric traits. We explore this proposition in the next chapter where we see that appropriately combining multiple biometric traits does significantly improve the performance of a biometric cryptosystem. However, we also identify some disadvantages of combining multiple traits, namely, the risk of loss of larger amount of biometric data if the template is compromised. We thus recommend some modifications to the basic framework in order to mitigate such fears.

Chapter 4

Multibiometric Cryptosystems

4.1 Introduction

A typical biometric system using only a single biometric trait generally suffers from limited recognition performance and substantial failure to enroll rate. To overcome these limitations, multibiometric systems are now prevalent. Multibiometric systems accumulate evidence from more than one biometric trait (e.g., face, fingerprint, and iris) in order to recognize a person [109] thereby leading to higher recognition accuracy and larger population coverage. Multibiometric systems are being widely adopted in many large-scale identification systems, including FBI's IAFIS, Department of Homeland Security's US-VISIT, and Government of India's UID. A number of software and hardware multibiometric products have also been introduced by biometric vendors [30, 84].

While multibiometric systems have improved the accuracy and reliability of biometric systems, sufficient attention has not been paid to security of multibiometric templates. Moreover, multibiometric templates contain information regarding multiple traits of the same user and are thus more attractive for adversaries. In this chapter we focus on the various aspects of designing a biometric cryptosystem that

can simultaneously secure multiple biometric traits represented in diverse forms.

Biometric cryptosystems have been originally designed only for specific biometric feature representations. For example, the fuzzy commitment scheme assumes a binary string representation, where the dissimilarity between template and query is measured in terms of the Hamming distance. The fuzzy vault assumes point-set based representations and uses set difference as the dissimilarity metric. Thus the unibiometric cryptosystems are not amenable to situations where multiple biometric traits of a user need to be encoded in a single biometric cryptosystem. Biometric traits are typically represented in three different forms: set of points, binary vector or a real valued vector. Point-set based features are used when the image has a set of salient points (e.g., fingerprint minutiae). If different samples of a biometric trait exhibit limited relative geometric transformation and limited occlusion, real-valued feature vectors obtained through PCA [128] and LDA [13] can be used. Binary strings are usually obtained through quantization of a real-valued feature vector, which reduces the storage space and matching complexity. For example, the bits in an iriscode [34] are obtained through quantization of the phase response of a Gabor filter applied to the corresponding iris image.

This diversity of biometric representations naturally requires a separate template protection scheme for each trait, and a fusion of the decisions made by each trait [49]. This is analogous to a security system that requires multiple low strength (fewer bits) passwords, which is less secure than a system that uses a single password with a larger number of bits. This motivates the proposed approach to protect the multiple biometric templates using a single secure sketch.

While the concept of securing multiple templates simultaneously as a single entity using a biometric cryptosystem has been reported in the literature, published approaches usually assume that different templates follow the same representation scheme. This enables simple concatenation of the individual templates to obtain the

fused template [70]. The objective of this work is to examine the feasibility of creating a single multibiometric secure sketch when the traits that are being fused have different feature representations.

4.2 Background

A number attempts have been made to extend the secure biometric recognition framework to incorporate multiple biometric traits [49, 70, 90, 119]. Sutcu et al. [119] combined face and fingerprint templates that are both transformed into binary strings. These binary strings are concatenated and used as the input to a fuzzy commitment scheme.

Nandakumar and Jain [90] proposed a multibiometric cryptosystem in which biometric templates based on binary strings and point-sets are combined. The binary string is divided into a number of segments and each segment is separately secured using a fuzzy commitment scheme. The keys associated with these segment-wise fuzzy commitment schemes are then used as additional points in the fuzzy vault constructed using the point-set based features.

Kelkboom et al. [70] provided results for feature level, score level and decision level fusion of templates represented as fixed-length real-valued vectors. Since the match scores are not explicitly available in a biometric cryptosystem, Kelkboom et al. used the number of errors corrected by an error correcting code in a biometric cryptosystem as a measure of the score. Such scores are, however, meaningful only if the crypto-biometric match is successful and the key κ_C can be successfully recovered. Moreover, multiple scores can be obtained only if the different templates are secured individually, which leads to suboptimal security. This is also true for decision level fusion. The feature level fusion scheme in [70] involves simple concatenation of two real-valued vectors and binarization of the combined vector using quantization thresholds.

Fu et al. [49] theoretically analyzed the template security and recognition accuracy imparted by a multibiometric cryptosystem, which can be operated in four different ways: no-split, MN-split, package, and biometric model. The first three models correspond to decision level fusion, where the biometric templates are secured individually. The biometric model is based on feature level fusion of homogeneous templates. However, no system implementation was reported.

Cimato et al. [29] follow a modular approach to design multibiometric cryptosystems. Suppose that $\mathbf{b}_1^E$ and $\mathbf{b}_2^E$ are two biometric templates. A secure sketch $\mathbf{y}_1$ is extracted from $\mathbf{b}_1^E$ along with a hash of the $\mathbf{b}_1^E$, which is further used as a key to secure the second template. This approach is similar to the package model proposed in [49], which in turn is based on the AND decision fusion rule. Fang et al. [37] consider a more general version of the above modular approach, where multiple secrets (could be biometric templates or passwords) are mixed in a cascaded fashion within the secure sketch framework. One advantage of such a modular approach is that additional biometric traits can be easily introduced in the multibiometric cryptosystem. Another benefit is that it allows the use of heterogeneous templates. For example, in [29], a secure sketch is used to protect the iriscode template, and the hash value of the iriscode based on the secret key is used to encrypt a fingerprint minutiae template. A limitation of this approach is that its overall security is bounded by the security of the sketch in the outermost layer.

Here, we propose a generic framework for the design of a multibiometric cryptosystem with heterogeneous templates and consider practical implementation issues in the case of both binary string and point-set based representations.

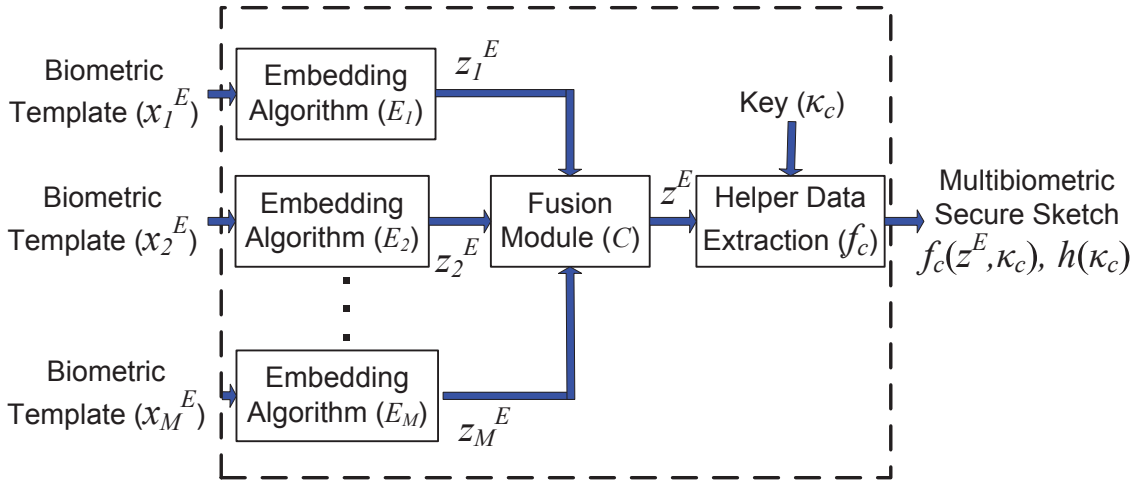


Figure 4.1: Schematic diagram of a multibiometric cryptosystem based on the proposed feature level fusion framework during the enrolment phase.

4.3 Multibiometric Cryptosystems Framework

We propose a feature level fusion framework for multibiometric cryptosystems that consists of three basic modules: (i) embedding algorithm ($\mathcal{E}$), (ii) fusion module ($\mathcal{C}$), and (iii) biometric cryptosystem ($\mathbf{f}_c$). The generic framework of the proposed multibiometric cryptosystem is shown in Figure 4.1. Suppose that we have a set of biometric feature representations $\boldsymbol{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M\}$, where $\mathbf{x}_m$ represents the features corresponding to the m^{th} biometric modality of a user, and M represents the number of modalities, $m = 1, 2, \cdots, M$. The functionalities of the three modules are as follows:

- **Embedding algorithm** ($\mathcal{E}$): The embedding algorithm transforms a biometric feature representation $\mathbf{x}_m$ into a new feature representation $\mathbf{z}_m$, where $\mathbf{z}_m = \mathcal{E}_m(\mathbf{x}_m)$, for all $m = 1, 2, \cdots, M$. The input representation $\mathbf{x}$ can be a real-valued feature vector, a binary string, or a point-set. The output representation $\mathbf{z}$ could be a binary string or a point-set that could be secured using fuzzy commitment or fuzzy vault, respectively.

- **Fusion module** ($\mathcal{C}$): The fusion module combines a set of homogeneous biometric features $\boldsymbol{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_M\}$ to generate a fused multibiometric feature representation $\mathbf{z}$. For point-set based representations, one can use $\mathbf{z} = \mathcal{C}_s(\boldsymbol{Z}) = \cup_{m=1}^{M} \mathbf{z}_m$. In the case of binary strings, the fused feature vector can be obtained by simply concatenating the individual strings, i.e., $\mathbf{z} = \mathcal{C}_b(\boldsymbol{Z}) = [\mathbf{z}_1\ \mathbf{z}_2\ \cdots\ \mathbf{z}_M]$. Note that it is also possible to define more complex fusion schemes, where features could be selected based on criteria such as reliability and discriminability.

- **Biometric cryptosystem** ($\mathbf{f}_c$): During enrolment, the biometric cryptosystem generates a secure sketch $\mathbf{y}_c$ using the fused feature vector $\mathbf{z}^E$ (obtained from the set of biometric templates $\boldsymbol{X}^E = \{\mathbf{x}_1^E, \mathbf{x}_2^E, \cdots, \mathbf{x}_M^E\}$) and a key κ_c, i.e., $\mathbf{y}_c = \mathbf{f}_c(\mathbf{z}^E, \kappa_c)$. During authentication, the biometric cryptosystem recovers κ_c from $\mathbf{y}_c$ and $\mathbf{z}^A$ (obtained from the set of biometric queries $\boldsymbol{X}^A = \{\mathbf{x}_1^A, \mathbf{x}_2^A, \cdots, \mathbf{x}_M^A\}$). Fuzzy commitment is used if $\mathbf{z}$ is a binary string, whereas a fuzzy vault is used if $\mathbf{z}$ is a point-set.

Each of the above three modules play a critical role in determining the matching performance and security of the multibiometric cryptosystem. The embedding algorithm should generate a compact representation that preserves the discriminability of the original biometric features. The fusion module should find the optimal trade-off between the discriminability and variability in the individual feature representations. The biometric cryptosystem should minimize the information leakage about the original biometric templates. Thus, optimizing each module is a challenging task in itself and is beyond the scope of this work. Since our primary objective is to demonstrate the viability of the proposed feature level fusion framework, we propose fairly simple algorithms for implementing the above three modules and do not focus on optimizing them.

4.3.1 Embedding Algorithms

We shall now discuss three types of embedding algorithms that can perform the following feature transformations: (i) real-valued vector into a binary string, (ii) point-set into a real vector, and (iii) binary string into a point-set (see Table 4.1).

Real-valued vector to binary string

A number of schemes have been proposed in literature for binarization of real-valued biometric features. Examples include Binary Multidimensional Scaling techniques [107], Locality Sensitive Hashing [11], Detection Rate Optimized Bit Allocation [27], and quantization of element pairs in the polar domain [26].

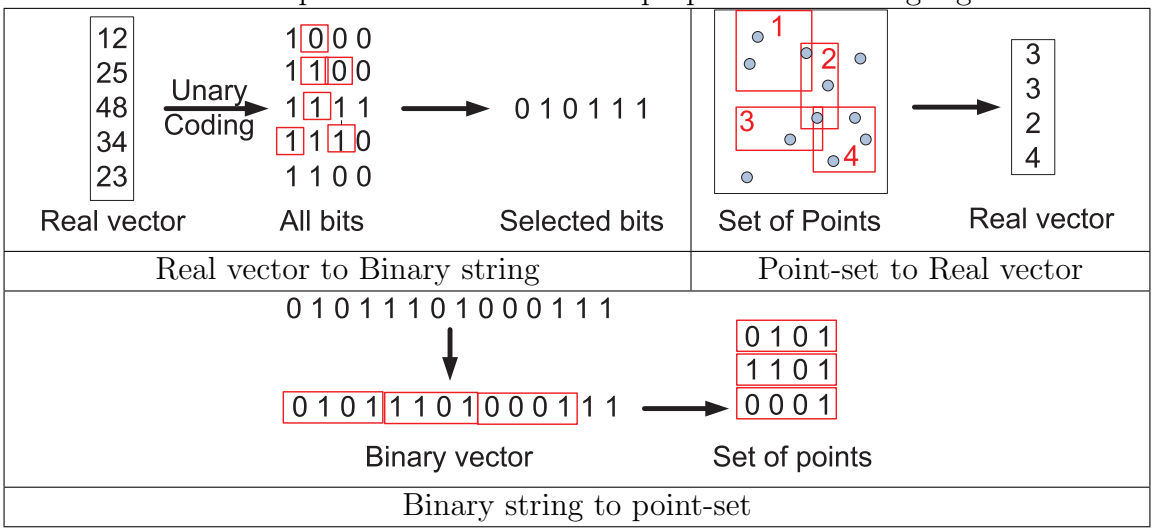
Since no single feature binarization technique is provably better than all others, we propose the following simple algorithm for transforming a real-valued vector into a binary string. First, we quantize each element of the real-valued vector into $(\tau + 1)$ fixed size quanta. The quantized values are then represented using τ-bit unary[1] representation in order to obtain a binary string of length $\tau\ell$, where ℓ is the dimensionality of the original vector. In the second stage, we select a desired number of most discriminable bits (N). The discriminability of each bit is computed as $((1 - p_g^e)p_i^e)$, where p_g^e and p_i^e are the genuine and impostor bit-error probabilities, respectively. Note that this procedure was also discussed in Section 3.5.1.

Point-sets to real vector

A number of techniques have been proposed for converting point-sets into binary feature vectors. These techniques include local point aggregates [88], spectral minutiae [136], geometric transformation [119], triplet histogram [38], and the bag-of-words

[1] A unary encoding works as follows. Suppose that a real-value a needs to be encoded using τ bits. The range of a, say $[a_{min}, a_{max}]$, is quantized into $(\tau+1)$ bins. If a falls into the i^{th} bin, it is represented as $(\tau - i + 1)$ ones followed by $(i - 1)$ zeros, where $i = 1, 2, \cdots, (\tau + 1)$.

Table 4.1: A simplified illustration of the proposed embedding algorithms.



approach [40]. In this paper, we implement the simple local aggregates based technique, which works as follows. Let us assume that each point can be represented as an ν-tuple. The available point-set is aligned such that the bounding box of the points is centered at the origin. Then, a set of axis-aligned hyper-rectangles with randomly selected position and size are generated. Among these hyper-rectangles, a fraction of hyper-rectangles with large overlap with other hyper-rectangles is discarded.

Statistics for each hyper-rectangle based on the points falling inside it are computed. These statistics include the number of points in the hyper-rectangle, and the mean and variance of the points along each of the ν dimensions. The statistics from different hyper-rectangles are concatenated to generate a feature vector. A Linear Discriminant Analysis (LDA) is applied to the resultant feature vector to reduce the dimensionality. Finally, the real-valued LDA features can be further binarized using the algorithm presented in Section 4.3.1.

Binary string to point-set

Conversion of binary string to point-set is required when the final biometric cryptosystem is based on point-set features. In order to obtain a point-set from a binary

string, we simply divide the binary string into the desired number of segments. Each segment can be considered as a point in the point-set representation. The only parameter in this technique is the number of segments. A similar technique was also used in [90], where instead of directly using the segments of the binary strings as points, a key is associated with each segment through fuzzy commitment and the keys are used as additional points in the vault.

4.3.2 Multibiometric Fuzzy Vault Implementation

A typical fuzzy vault is constructed as a set of 3-tuples $\mathbf{y}_c = \{(\alpha_i, \beta_i, \gamma_i)\}_{i=1}^{t}$, where $t = (r+q)$, $\alpha_i \in (\mathbf{s}^E \cup \mathbf{s}^C)$, β_i is the points in $\mathcal{F}$ associated with points in $(\mathbf{s}^E \cup \mathbf{s}^C)$, and γ_i is given by

$$\gamma_i = \begin{cases} P(\beta_i), & \text{if } \alpha_i \in \mathbf{s}^E, \\ \\ b_i, \text{where } b_i \in \mathcal{F} \setminus \{P(\beta_i)\}, & \text{if } \alpha_i \in \mathbf{s}^C. \end{cases} \tag{4.1}$$

Here, $\mathbf{s}^E$ is a set based biometric template, $\mathbf{s}^C$ is the set of chaff points. See Section 3.3.1 for further details. During decoding, the points in the vault are ordered based on their likelihood of being genuine which is estimated by matching the points in the vault with the query. Given the ordered set of points, the vault is decoded according to Algorithm 3.1.

Multiple unibiometric vaults can be easily converted into a single multibiometric vault by associating the same key κ_c with them. The key length (L) and hence, the polynomial degree k of a multibiometric vault is typically set to a higher value compared to the unibiometric case. During decoding, multiple query biometrics are matched with the corresponding unibiometric vaults and an ordered sequence of points from each vault is obtained. These individual sequences of points are then merged such that the first l elements of the merged sequence contain approximately top $\eta_i l$

points from the vault corresponding to the ith biometric. In the current implementation, we choose η_i to be the same for all the biometric traits. However, specific strategies can be designed to select proper values of η_i based on the quality of the individual biometric traits and the number of genuine points from each trait.

4.3.3 Multibiometric Fuzzy Commitment Implementation

In the fuzzy commitment technique, the biometric template $\mathbf{b}^E$ of length N is bound to a codeword $\mathbf{c}$ of the same length to generate the secure sketch $\mathbf{y}_c = \mathbf{b}^E \oplus \mathbf{c}$. During authentication, the query biometric data, $\mathbf{b}^A$, is XOR'ed with the secure sketch to obtain a corrupted codeword $\mathbf{c}^*$, which can be corrected to recover the key κ_c that is associated with the codeword $\mathbf{c}$. See Section 3.3.2 for further details about the decoding procedure.

In order to create a multibiometric cryptosystem with M different biometric traits, we extract $N = 1,023 \times M$ most discriminative bits from the pool of bits available from all the constituent biometric traits. In our experiments, we assume different values of D_{min} (the minimum distance of the error correcting code) in the range 0.02 to 0.6 times the total number of bits N.

4.3.4 Constrained Multibiometric Cryptosystem

One of the limitations of a multibiometric system is that it is possible for an adversary to get successfully authenticated by spoofing only a subset of the involved biometric traits [108]. This issue is also a concern for a multibiometric cryptosystem. Ideally, a multibiometric system should ensure the presence of a minimum amount of discriminatory information from a subset or all the biometric traits of the user, especially those that are difficult to spoof. We refer to a cryptosystem that enforces such a requirement as a *constrained multibiometric cryptosystem* and the traits for which a minimum matching constraint is applied as *constrained traits*.

There are many ways to impose a minimum matching constraint for a biometric modality within a multibiometric cryptosystem. For example, when only two modalities are involved, it is possible to set the error correction capacity in such a way that even a perfect match in one modality is not sufficient to decode the secure sketch and some minimum level of similarity is also required for the second modality. Such an approach will have high template security, but will reduce the GAR significantly. Alternatively, one can store separate unibiometric sketches for each modality and allow them to be decoded individually. This approach will lower the security, but will result in higher GAR compared to the first approach.

We propose a constrained multibiometric cryptosystem that does not affect the security of a multibiometric secure sketch, but enforces a matching constraint on individual modalities. Our approach is conceptually similar to the modular multibiometric cryptosystem proposed in [29]. The proposed approach assumes that two different representations called the *primary* and *secondary* representations are available for the constrained biometric modalities. These two representations satisfy the following property: it should be hard to obtain the *primary* representation from the *secondary* representation. A simple way to satisfy this requirement is to consider the given biometric feature vector (e.g., minutiae set) as a primary representation and derive the secondary representation by applying a non-invertible transformation (e.g., minutiae aggregates [88]) to the given feature vector. Thus, even if the secondary representation is revealed, it is difficult to obtain the primary representation.

For each of the constrained trait, its *secondary* representation is secured using the multibiometric cryptosystem using the feature level fusion framework whereas its *primary* representation is secured using a unibiometric cryptosystem (see Figure 4.2). The unibiometric cryptosystems corresponding to the various constrained traits will use unique keys that are different from the one used in the multibiometric cryptosystem. Finally, the unibiometric secure sketches are encrypted with a symmetric

cryptographic algorithm such as AES (Advanced Encryption Standard) [6], where the encryption key is the same as the key associated with the multibiometric cryptosystem. The authentication involves two stages. In the first stage, the key associated with the multibiometric cryptosystem is recovered. This key is used to decrypt the unibiometric secure sketches. In the second stage, the unibiometric secure sketches are decoded. All the keys associated with the unibiometric sketches must be correctly recovered for successful authentication.

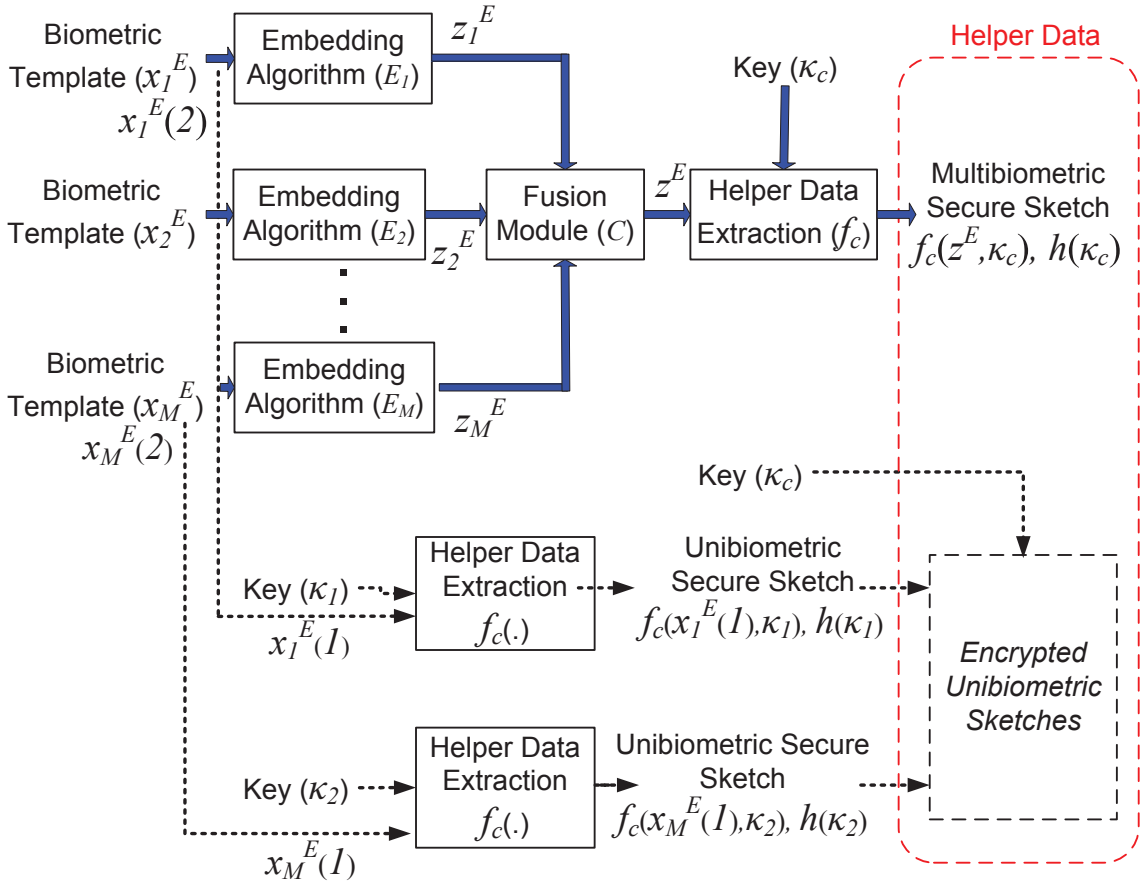


Figure 4.2: Enrolment phase of a constrained multibiometric cryptosystem. The templates corresponding to each constrained trait (traits 1 and M in this example) have two representations (the primary representation ($\mathbf{x}_i^E(1)$) and the secondary representation ($\mathbf{x}_i^E(2)$) for modality i). The secondary representation is secured using a multibiometric secure sketch, while the primary representation is secured using a unibiometric sketch that is further encrypted using the key associated with the multibiometric cryptosystem.

Unlike the simple multibiometric cryptosystem shown in Figure 4.1, the constrained multibiometric cryptosystem requires storage of both multibiometric and unibiometric secure sketches. But the proposed approach has two advantages. Firstly, the overall security of the templates is not affected because unibiometric sketches are encrypted using the key that is bound to the multibiometric sketch; unless the attacker decodes the multibiometric sketch he cannot compromise the unibiometric sketches. Secondly, the *primary* representation that is required to decode a unibiometric sketch cannot be obtained from the *secondary* representation. But successful authentication requires decoding of the multibiometric sketch as well as all the unibiometric sketches. This ensures that the user has a minimum amount of information about each of the constrained biometric traits. The limitation of the proposed approach is that it leads to a degradation in the GAR because it is possible that an authentication attempt fails despite correct decoding of the multibiometric sketch, because one or more of the unibiometric sketches may not be decoded correctly.

4.4 Experimental results

The experimental set-up of this chapter follows the set-up described in Chapter 3. We use the same three databases in our evaluation, namely, the Fingerprint Verification Competition (FVC) 2002 Database-2, the CASIA Iris database Ver-1, the XM2VTS face database, and the West Virginia University (WVU) multimodal database containing fingerprints, iris and face images. Same as in Chapter 3, we consider one genuine authentication attempt per user and impostor attempts are simulated by using one impression of each user's biometric to authenticate as every other user. Consequently, the number of impostor attempts N_I is $9,900$ (100×99) for the virtual multimodal database and $18,906$ (138×137) for the real multimodal database. In order to evaluate both fuzzy vault as well as fuzzy commitment, we obtain both

point set as well as binary vector based features from each of the three biometric traits, namely, fingerprint, iris, and face.

A point set based representation for fingerprints and a binary vector based representation for iris and face have been detailed in Section 3.5. In order to extract a binary vector from fingerprints, we follow the approach outlined in Section 4.3.1 with 500 hyper-rectangles (cuboids in 3D space) aligned along the horizontal location, vertical location, and orientation axis associated with minutiae. Different features such as sum of distances of minutiae from the six walls of the cuboids and mean and standard deviations of minutiae along each of the three axes, are extracted from each cuboid in order to obtain a vector of length $3,500$. Linear Discriminant Analysis (LDA) is used to reduce the dimensionality of this vector to 80. Each LDA coefficient is converted into a 40-bit unary representation and they are concatenated to obtain a $3200(= 40 \times 80)$-bit binary string. We select a subset of the most discriminable bits (N_p) using the procedure described in Section 4.3.1. First impression of the finger is used for enrolment, the second one is used as authentication sample and the remaining impressions are used as training set in order to compute the LDA features. Since no training is required for extracting minutiae, only the first two impressions are used in constructing the fuzzy vault. To obtain the point-set representation from iris and face, 800 bits selected from the binarized LDA features are divided into 20 segments of 40-bits each.

Figures 4.3 and 4.4 show the performance of the multibiometric fuzzy vault for the virtual and real multimodal databases, respectively. In general, it can be observed that incorporating additional biometric features does increase the performance of the system. In case of the virtual multimodal database, the security of the iris fuzzy vault at a GAR of 90% is 45 bits; however, when fingerprint and face are also incorporated in the fuzzy vault, the security increases to around 90 bits at the same GAR. When the templates are secured individually and the AND fusion rule is applied, i.e., the

authentication is deemed successful only when all the unibiometric cryptosystems are decoded, the security at 90% GAR is around 40 bits. However, in case of the WVU database, there is only a marginal increase in performance compared to the best modality (face). This can be attributed to the lower quality of the iris and fingerprint images in the WVU database compared to the CASIA and FVC2002-DB2 databases, respectively. In fact, the GAR of the iris fuzzy vault for the WVU database at zero-FAR is 0%, which is the reason why the G-S curve corresponding to iris is not shown in Figure 4.4. Note, it is possible that a poor quality sample from one of the modalities can lead to a higher decoding complexity if the relative quality of the samples is not taken into account when generating the multibiometric template. In order to address this issue, we also check if any subset of biometric modalities can decode the vault. The final value of security is the minimum among the security based on the multibiometric query and that based on different subsets of the query biometric traits.

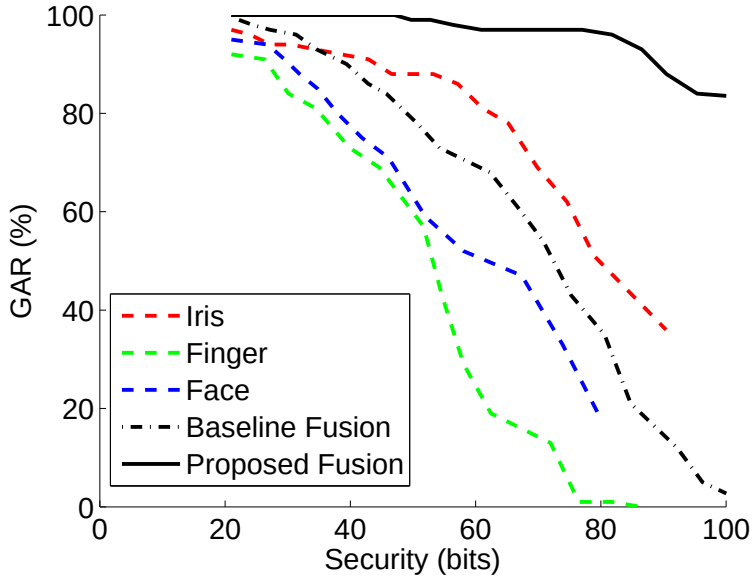


Figure 4.3: The G-S curves for fuzzy vault for iris, fingerprint, and face images from CASIA Ver-1, FVC 2002 DB-2, and XM2VTS databases, respectively, the baseline multibiometric cryptosystem based on AND-fusion rule and the proposed multibiometric crytposystem using all three modalities.

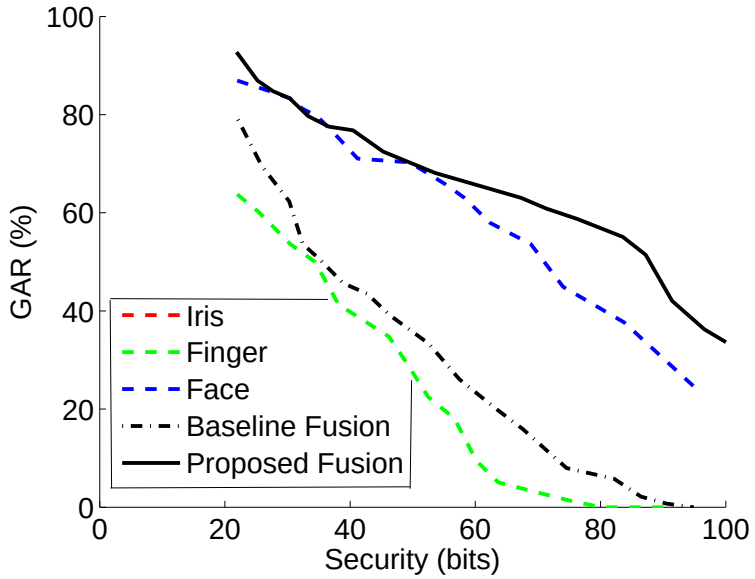


Figure 4.4: The G-S curves for fuzzy vault for iris, fingerprint, and face images from WVU Multimodal database, the baseline multibiometric cryptosystem based on AND-fusion rule and the proposed multibiometric crytposystem using all three modalities.

The results corresponding to fuzzy commitment are shown in Figures 4.5 and 4.6 for the virtual and real multimodal databases, respectively. The G-S curves are obtained by varying D_{min} of the error correcting code. Similar to fuzzy vault, the performance of the multibiometric fuzzy commitment is significantly better than the unibiometric systems.

Table 4.2 summarizes the GAR of different biometric cryptosystems at a security level of 53 bits (equivalent of 8-character password randomly chosen from a 94-character alphabet). We observe that the performances of the unibiometric cryptosystems are quite low, which may be due to three reasons. Firstly, as mentioned earlier, the quality of iris and fingerprint samples in the WVU multimodal database is substantially lower than the quality of samples in the FVC2002-DB-2 and CASIA ver1 databases, respectively. This explains the inferior performance of iris and fingerprint-based cryptosystems when evaluated on the WVU multimodal database. Secondly,

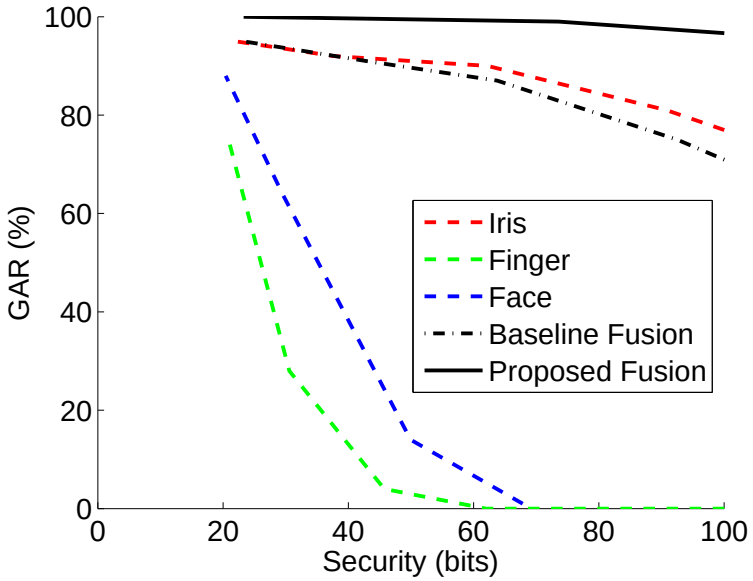


Figure 4.5: The G-S curves for fuzzy commitment for iris, fingerprint, and face images from CASIA Ver-1, FVC 2002 DB-2, and XM2VTS databases, respectively, the baseline multibiometric cryptosystem based on AND-fusion rule and the proposed multibiometric crytposystem using all three modalities.

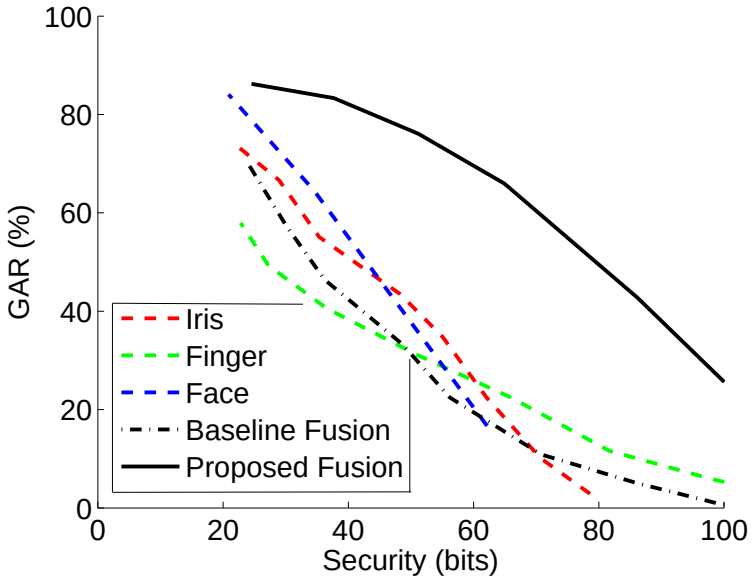


Figure 4.6: The G-S curves for fuzzy commitment for iris, fingerprint, and face images from WVU Multimodal database, the baseline multibiometric cryptosystem based on AND-fusion rule and the proposed multibiometric crytposystem using all three modalities.

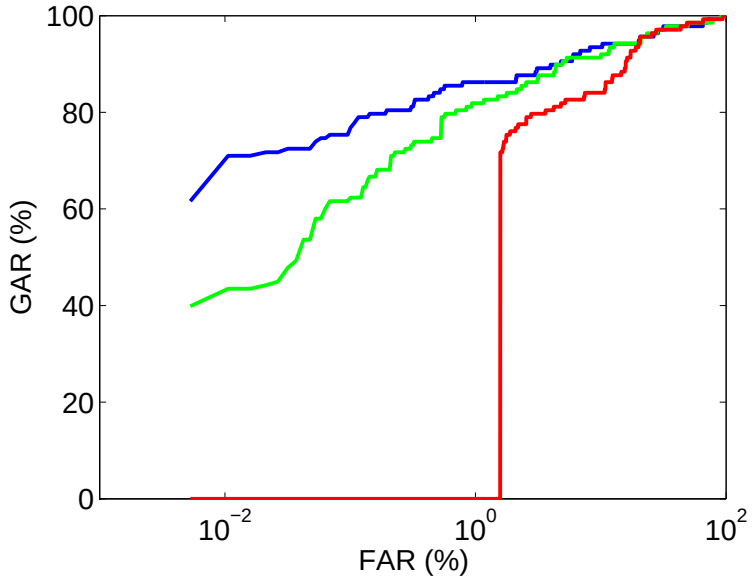
Traits	Real Multimodal Database		Virtual Multimodal Database	
	Fuzzy vault	Fuzzy commitment	Fuzzy vault	Fuzzy commitment
Iris	0%	37%	88%	91%
Finger	22%	30%	51%	2%
Face	67%	33%	58%	12%
Baseline Fusion	33%	27%	75%	89%
Proposed Fusion	68%	75%	99%	99%

Table 4.2: Comparison of genuine accept rates of the different biometric cryptosystems at a security level of 53 bits, which equals the security imparted by a randomly chosen 8 character password [22]. Here, baseline fusion refers to securing individual templates using unibiometric cryptosystems and combining decisions using AND-rule fusion, while the proposed fusion scheme uses a single multibiometric secure sketch.

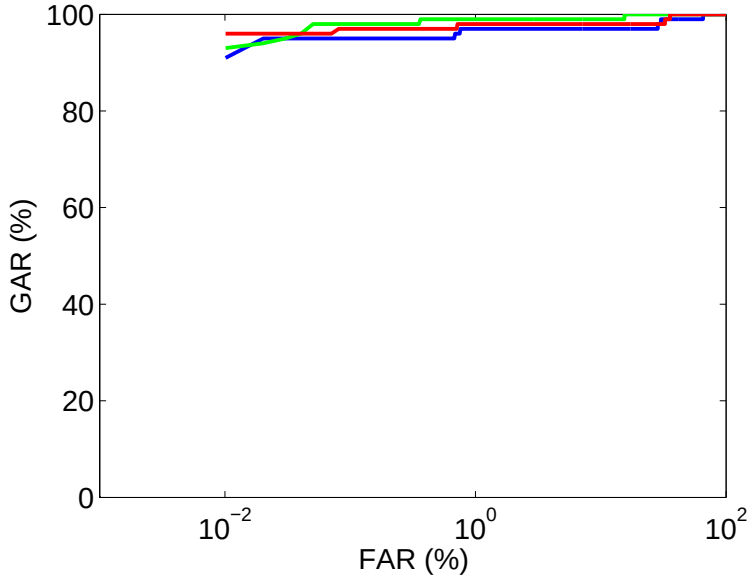
there is a loss of discriminatory information during the feature transformation (embedding) stage. See Figures 4.7, 4.8, and 4.9. This explains the better performance of the unibiometric cryptosystems when the native representation scheme is used. For example, in both the real and virtual multimodal databases, iris fuzzy commitment performs better than a iris fuzzy vault. Similarly, the performance of fingerprint fuzzy vault is generally better than a fingerprint fuzzy commitment.

For the multibiometric fuzzy vault implementation reported in [90], where iris and fingerprint templates from MSU-DBI database and CASIA Ver-1 database, respectively, were secured together, the genuine accept rate was 98.2% at a security of 49 bits. Note that the security estimate in [90] assumes uniform distribution of biometric features. In our implementation, the genuine accept rate is 99% at a security of 49 bits based on the FVC2002-DB2 and the CASIA Ver-1 databases. In [29], security of the system has not been explicitly reported. In [70], the proposed technique performs fusion of two different 3D face recognition algorithms and thus cannot be directly compared to the techniques proposed here. In [49], no experimental results were reported.

To validate the constrained multibiometric cryptosystem, we implemented a sys-

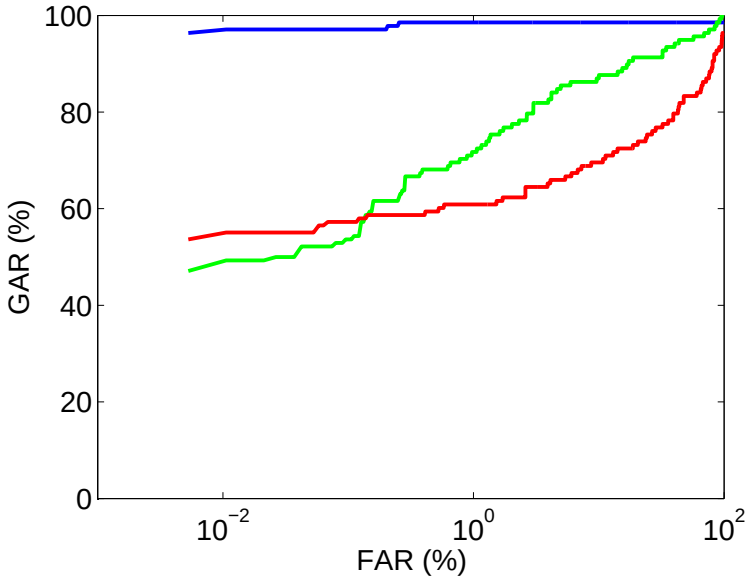


(a)

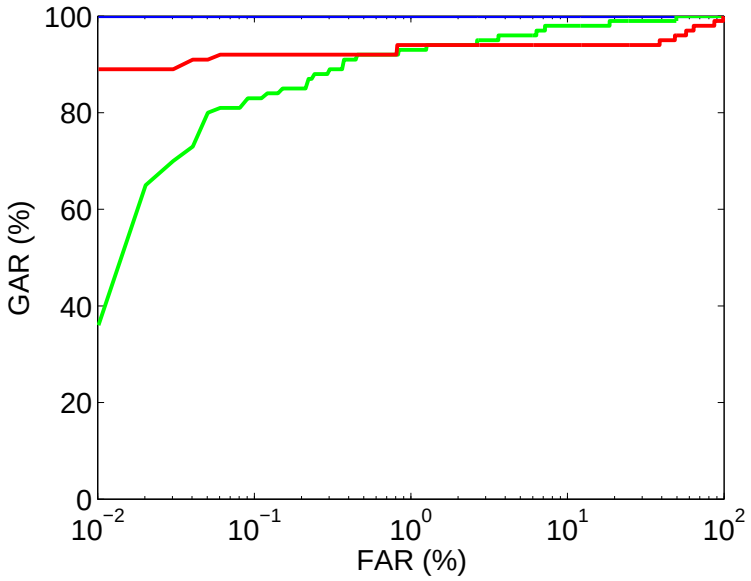


(b)

Figure 4.7: ROC curves corresponding to the original features (blue), features processed for fuzzy commitment (green) and features processed for fuzzy vault (red) for Iris images from (a) WVU and (b) CASIA Ver-1 databases. The ROC curves corresponding to the original features is based on the Hamming distance between iriscodes. The curves corresponding to the fuzzy commitment are based on Hamming distance between $1,023$ bits of the extracted binary feature vector.

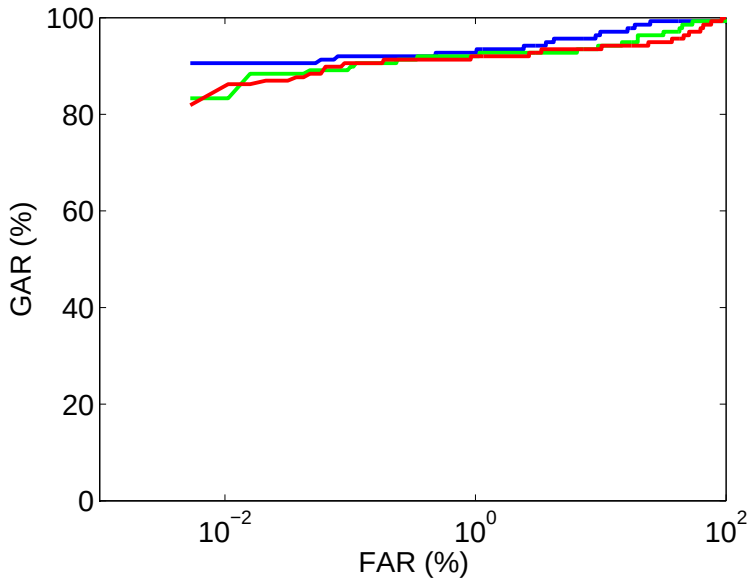


(a)

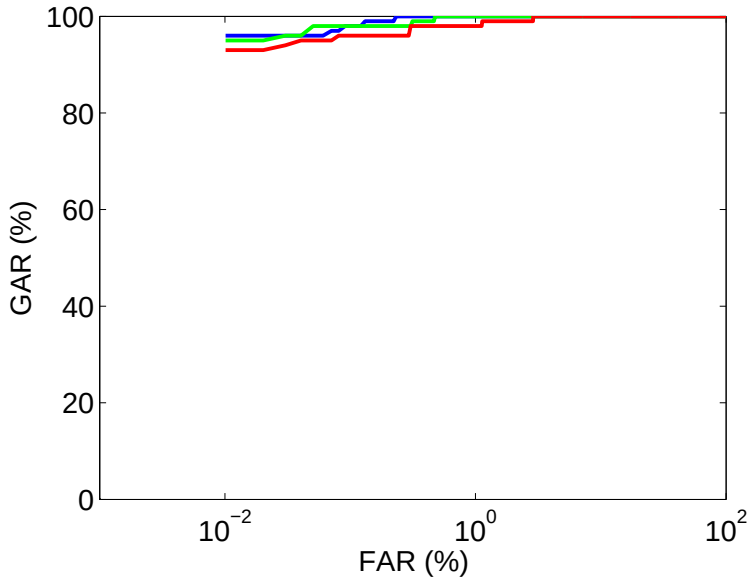


(b)

Figure 4.8: ROC curves corresponding to the original features (blue), features processed for fuzzy commitment (green) and features processed for fuzzy vault (red) for fingerprint images from (a) WVU and (b) FVC02DB2 databases. The ROC curves corresponding to the original features is based on the scores obtained from Neurotechnology Verifinger matcher using only the minutiae features. The curves corresponding to the fuzzy vault are computed using the decoding complexity as the matching score when a degree-10 polynomial used.



(a)



(b)

Figure 4.9: ROC curves corresponding to the original features (blue), features processed for fuzzy commitment (green) and features processed for fuzzy vault (red) for face images from (a) WVU and (b) XM2VTS databases. The ROC curves corresponding to the original features is based on the LDA features. The curves corresponding to the fuzzy commitment are based on Hamming distance between $1,023$ bits of the extracted binary feature vector.

tem consisting of iris and fingerprint modalities, where minimum matching constraints are imposed for the fingerprint modality. We further assume that the adversary has knowledge about iris biometric, i.e., he has access to some iris image of the enrolled user. In this experiment, a multibiometric fuzzy commitment is implemented and a *secondary* representation of fingerprints is obtained using minutiae aggregates. Minutiae are employed as the *primary* fingerprint representation, and hence a fuzzy vault is used in the second stage. The degree of polynomial for the fuzzy vault is selected such that the sum of security in bits and GAR in percentage of the resulting system is maximized. Using this constrained multibiometric cryptosystem, it is possible to achieve a security of 35 bits even if the iris features of a genuine user are known the adversary. However, the GAR for this scenario is only 15% compared to a GAR of 70%, when no constraints were imposed on the fingerprint modality.

4.5 Summary

We have proposed a feature-level fusion framework for the design of multibiometric cryptosystems that simultaneously protects the multiple templates of a user using a single secure sketch. The feasibility of such a framework has been demonstrated using both fuzzy vault and fuzzy commitment, which are two of the most well-known biometric cryptosystems. We have also proposed different embedding algorithms for transforming biometric representations, efficient decoding strategies for fuzzy vault and fuzzy commitment, and a mechanism to impose constraints such as minimum matching requirement for specific modalities in a multibiometric cryptosystem. A realistic security analysis of the multibiometric cryptosystems has also been conducted. Experiments on two different multibiometric databases containing fingerprint, face, and iris modalities demonstrate that it is indeed possible to improve both the matching performance and template security using the multibiometric cryptosystems. We

also noted that the matching performance is noticeably degraded when biometric cryptosystem is used compared to the case unsecured templates are matched. Also, matching performance may vary significantly based on the type of embedding algorithm and biometric cryptosystem used. In general, the matching performance is expected to be higher if the biometric cryptosystem is applied on the native representation of a biometric trait compared to the case when an embedding algorithm is applied to transform the biometric trait to a different representation.

Chapter 5

Augmented Fingerprint Vault

5.1 Introduction

Fingerprint fuzzy vault is one of the most commonly studied biometric cryptosystems due to its ability to secure fingerprint templates represented in the form of a set of minutiae. Since it was first proposed in 2002 by Juels and Sudan [68], a number of improvements have been made to its basic construction but still a number of limitations need to be overcome before fuzzy vault can be practically viable. In this chapter we study two main limitations of a fingerprint fuzzy vault. The first limitation is that it is indeed possible to determine that two different fuzzy vaults were constructed from the same finger. Note that in a fuzzy vault, minutiae are obscured with a large number of randomly generated points. While it is difficult to identify the true minutiae among the chaff points in a single fuzzy vault, if two fuzzy vaults obtained from the same finger are overlaid on each other, the genuine points that are common between the two vaults can be identified. See e.g. [112]. In this chapter, we introduce user password for transforming the minutiae such that it is difficult to correlate two so called password-hardened vaults constructed using the same finger but with different passwords.

The second limitation of a fuzzy vault is that only simple features such as a set of points can be secured using the fuzzy vault framework. This severely restricts the performance of the fuzzy vault since a typical fingerprint matching system utilizing features in addition to minutiae leads to much greater performance than minutiae matching alone. See, for example, [41] where minutia descriptors were used to design a state of the art fingerprint matcher. In this chapter we discuss a technique to effectively incorporate minutiae descriptors into a fingerprint fuzzy vault that leads to significantly improved performance of the fuzzy vault.

Section 5.2 describes the technique used to incorporate the passwords into a fingerprint fuzzy vault framework whereas Section 5.3 presents the technique to incorporate the neighborhood information of a minutia in the fuzzy vault to improve its performance.

5.2 Fingerprint Vault with Passwords

To incorporate passwords into a fingerprint fuzzy vault, a random transformation function derived from the user's password is applied to the biometric template consisting of a set of minutiae. The transformed template is then secured using the fuzzy vault framework. Since it is not straight forward to match two templates obtained from the same user using different passwords, this scheme prevents linkage of templates across different applications and also allows re-issuance of new templates in case one is compromised. Moreover, as a result of the added randomness offered by the password, the distribution of the set of points in the template approaches a uniform distribution after transformation[1] and thus decreases the similarity between the transformed templates of different users. This also provides better resistance against attacks on the vault where the attacker tries to find genuine minutiae in the vault.

[1]Note that the distribution of minutiae is known to be non-uniform. See [141] for details.

5.2.1 Minutiae Transformation using Passwords

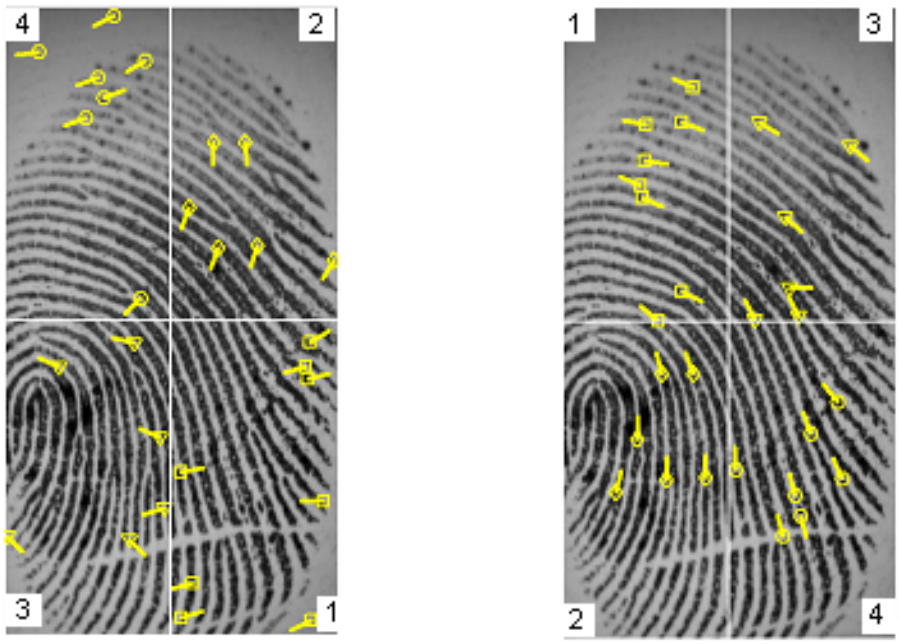
In order to transform a set of minutiae using a password, we partition the minutiae based on the quadrant of the image they lie into four groups and assign a 16 bit number obtained from the password to each group. We assume that the password is of length 64 bits (8 characters) which is divided into 4 units of 16 bits each. Each password unit is used to transform minutiae in one of the four quadrants of the fingerprint image. We quantize each minutia into a 16 bit number which can be considered as a point from the finite field $GF(2^{16})$. For this, each of the x, y, and θ components of a minutia representation are quantized into 2^{B_u}, 2^{B_v}, and 2^{B_θ} bins, respectively to obtain their quantized counterparts, namely, Q_u, Q_v, and Q_θ. Here, the value of $(B_u + B_v + B_\theta)$ is 16. The 16 bit password unit is also similarly divided into three components T_u, T_v and T_θ of lengths B_u, B_v and B_θ bits, respectively. The components T_u and T_v are considered as the binary representations of the amount of translations along the vertical and horizontal directions, respectively, and T_θ is treated as the binary representation of the change in minutia orientation. The new minutiae attributes are obtained by adding the translation values to the original values modulo the appropriate range, i.e.,

$$Q_u = Q_u + T_u \pmod{2^{B_u}} \tag{5.1}$$

$$Q_v = Q_v + T_v \pmod{2^{B_v}} \tag{5.2}$$

$$Q_\theta = Q_\theta + T_\theta \pmod{2^{B_\theta}} \tag{5.3}$$

Note that minutiae translation does not affect the intra-user variability of the minutiae features thereby maintaining the false reject rate to a great extent. There is some difference due to the boundary effects though. In order to further increase the randomness of the transformed template, we generate a permutation sequence of 4



(a) (b)

Figure 5.1: Minutiae transformation using password. (a) Original minutia distribution and (b) distribution of minutiae after password based transformation is applied.

numbers by applying a one way function[2] on the password. Using this sequence, we permute the 4 quadrants of the image such that the relative positions of minutiae within each quadrant are not changed. The effect of minutiae transformation using password is shown in Figure 5.1.

Apart from the well-known factors like partial overlap, non-linear distortion and noise that lead to differences between the template and query minutiae sets of the same user, the password-based transformation scheme introduces some additional discrepancies. If a minutia lies close to the quadrant boundary, the same minutiae may fall in different quadrants in the template and the query due to imperfect alignment. This reduces the number of minutiae correspondences and may lead to a small decrease in the genuine accept rate. Another problem arising due to imperfect alignment

[2] A function is considered to be one way if it is computationally easy to compute the function but it is computationally or information theoretically hard to recover the pre-image of a value with respect to this function

between query and template is that the same minutia point may appear at opposite ends of the quadrants in the template and the query after the transformation. This is because the minutiae are translated within their respective quadrants modulo the quadrant size. To address this problem, we add a border of width 15 pixels around each quadrant and minutiae that lie within 15 pixels of the quadrant boundary are duplicated on the border at the opposite end of the quadrant.

5.2.2 Experiments

The transformed minutiae are encoded in a vault using the same procedure as that used for a fingerprint. See Section 3.2 for further details. For the sake of simplicity, our analysis here assumes use of Lagrange interpolation based decoding procedure described in Section 3.3.1. Recall that in a Lagrange interpolation based decoding, all possible sets of points among the overlapping set with cardinality $k + 1$ are used to reconstruct the polynomial. Using this procedure, the decoding will be successful as long as more than k points are correctly identified in the set of candidate genuine points, where k is the degree of the secure polynomial. The complexity of this decoding depends on the number of chaff points in the vault identified as genuine.

The proposed password-based fuzzy vault hardening scheme has been tested on the FVC2002-DB2 database. Only the first two impressions of each of the 100 different fingers were used in our experiments; the first impression was used as the template to encode the vault and the second impression was used as the query in vault decoding. Here, the criteria used for evaluating the performance are failure to capture rate (FTCR), genuine accept rate (GAR) and false accept rate (FAR). When the number of minutiae in the template and/or query fingerprint is less than the required number of genuine points, we call it as failure to capture.

The security of the password augmented fuzzy vault is at least as good as the security of the original fuzzy vault. The security of the basic fuzzy vault is computed

based on a brute-force attack by trying to decode the vault using all possible combinations of $(k+1)$ points in the vault. If $k = 10$, $r = 30$ and $s = 300$, the total number of possible combinations is $\binom{330}{11}$. See Section 3.4.1 for details. Among these combinations, $\binom{30}{11}$ combinations will successfully decode the vault. The expected number of combinations that need to be evaluated is thus 2×10^{12} which corresponds to around 40 bits of security. Security can be improved by adding a larger number of chaff points (e.g., when $s = 600$ in the above system, we can achieve around 50 bits of security) at the expense of increased storage requirements and slight decrease in the GAR of the system. In order to further improve the security, the user's password can be used to encrypt the vault. Thus, during authentication, the user has to first decrypt the vault using his password before he can decode the vault.

Table 5.1 shows that the proposed system leads to a small decrease in the GAR for all values of k. This is due to different bin placement of a few minutiae at the quadrant boundaries and the inability of the minutiae matcher to effectively account for non-linear deformation in the transformed minutiae space.

In case the attacker knows the biometric template (e.g., by lifting a fingerprint impression left by the genuine user), he still needs to guess the password which is required to decrypt the fuzzy vault before any vault decoding can be performed. Note that although the maximum value of security imparted by an 8 character password is 53 bits, due to the predictable manner in which a user selects a password, the

	FTCR	k=7		k=8		k=10	
		GAR	**FAR**	**GAR**	**FAR**	**GAR**	**FAR**
Vault without hardening	2%	91%	0.13%	91%	0.01%	86%	0%
Hardened vault	2%	90%	0%	88%	0%	81%	0%

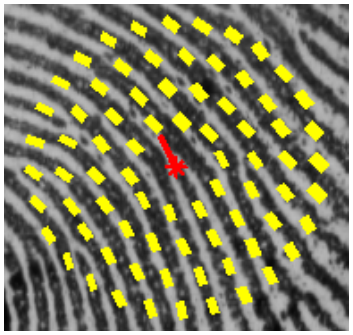
Table 5.1: Genuine Accept Rates (GAR), False Accept Rates (FAR) and Failure to Capture Rates (FTCR) of the hardened fuzzy vault for FVC2002-DB2 database. Here, k represents the degree of the polynomial used in vault encoding.

security imparted by an 8 character password can be as low as 18 bits [22]. When an adversary does not have any knowledge of the user password and user biometric data, then the security of the hardened fuzzy vault is a combination of the security provided by the password and biometric layers. If $k = 10$, $r = 30$, $s = 300$ and the password is 8 character long, the security of the hardened vault is ~ 58 bits. It is, however, important to note here that the measures of security and GAR provided here are not directly comparable to the security and accuracy provided in Chapters 3 and 4 due to the difference in the decoding procedure utilized as well as the assumptions about the capability of the attacker. Nevertheless, the analysis here provides a clear indication of the advantage of incorporating passwords into the basic construction of fingerprint fuzzy vault.

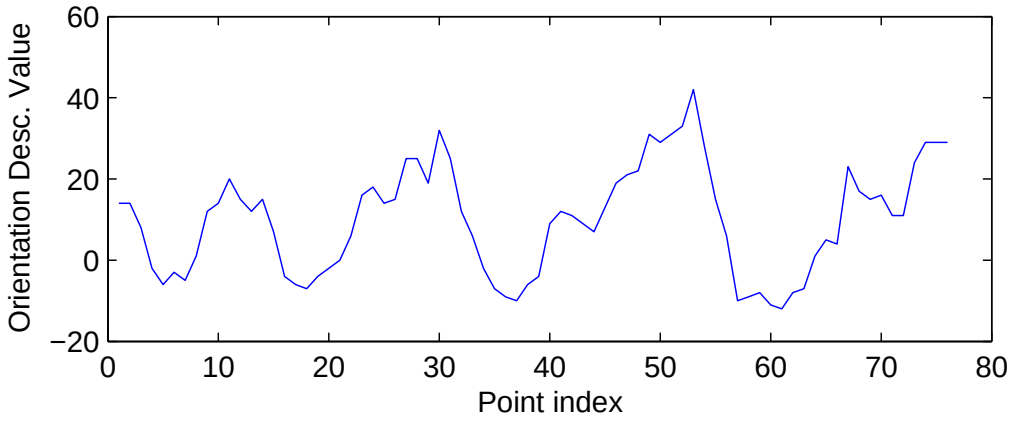
5.3 Fingerprint Vault with Minutiae Descriptors

Another way to improve the performance of a fingerprint fuzzy vault is by incorporating additional attributes extracted from a minutia's neighborhood into the vault. In particular, we use minutiae descriptors [41] that contain local ridge orientation and ridge frequency information, and show that they have sufficient saliency to improve the security (by reducing the FAR) of a fingerprint fuzzy vault.

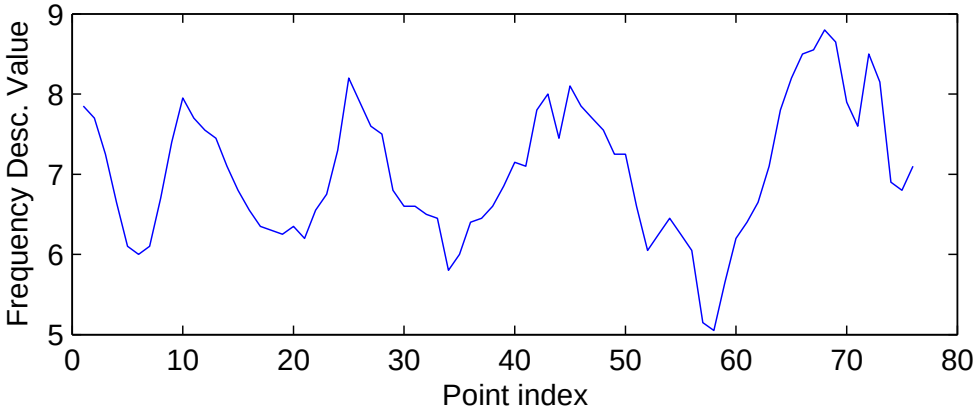
The minutia descriptor used here consists of texture based features in the form of ridge orientation and frequency values sampled at 76 equidistant points, uniformly spaced on 4 concentric circles around a minutia. The four concentric circles, with radius 27, 45, 63 and 81 pixels, contain 10,16, 22 and 28 points, respectively (see Figure 5.2). This configuration of points is based on the criteria that the difference between radii of two consecutive concentric circles and that between two sampled points on a circle should be twice the ridge period. Sampling the points in this manner captures maximum information contained in the neighborhood of a minutia [126].



(a)



(b)



(c)

Figure 5.2: Minutiae descriptor: (a) positions of 76 points in the neighborhood of a minutiae; thickness of each line and its orientation corresponds to frequency and orientation descriptors, (b) orientation descriptor and (c) frequency descriptor.

In order to effectively incorporate the discriminative information of minutiae descriptors in the fuzzy vault, we "encrypt" the ordinate values corresponding to a minutia using the descriptor associated with it[3]. To account for the noise in the measurement of the minutiae descriptor, a fuzzy commitment scheme is used for securing the ordinate values instead of the traditional cryptographic techniques such as the Advanced Encryption Standard (AES). Since the descriptor corresponding to a minutia is more likely to decode the associated ordinate value during a genuine match than during an impostor match, the proposed hybrid cryptosystem improves the matching performance as well as the security of the vault.

5.3.1 Vault Encoding/Decoding

In the proposed fingerprint cryptosystem, vault construction consists of two main steps : (i) fuzzy vault encoding, and (ii) securing ordinate values (see Figure 5.3). The vault encoding follows the same procedure as described in Section 3.3.1 whereas the procedure to secure the ordinate values is described below.

Once the basic fingerprint fuzzy vault is constructed, the ordinate values of the vault are secured using the fuzzy commitment approach where the biometric information involved comes from the binary strings extracted from the minutiae descriptors. The minutiae descriptors are binarized using the procedure described in Section 5.3.2. Let $D_i^b, i = 1, \cdots, (r+q)$ be the binary descriptor and C_i be a codeword generated from the corresponding 16 bit ordinate value γ_i. Instead of γ_i, only the secure ordinate value i.e. $G_i (= (D_i^b \oplus C_i))$ is stored in the vault as the fuzzy commitment. Here, the descriptors for the chaff points are chosen at random from the set of all the descriptors in the database. The set of abscissa values, the set of secure ordinate

[3]Note that storing the descriptors along with minutiae in the vault is not recommended as the descriptors can be used to verify whether two neighboring minutiae belong to the same fingerprint or not. This fact can be leveraged by the adversary to speed up the search for genuine points in the vault.

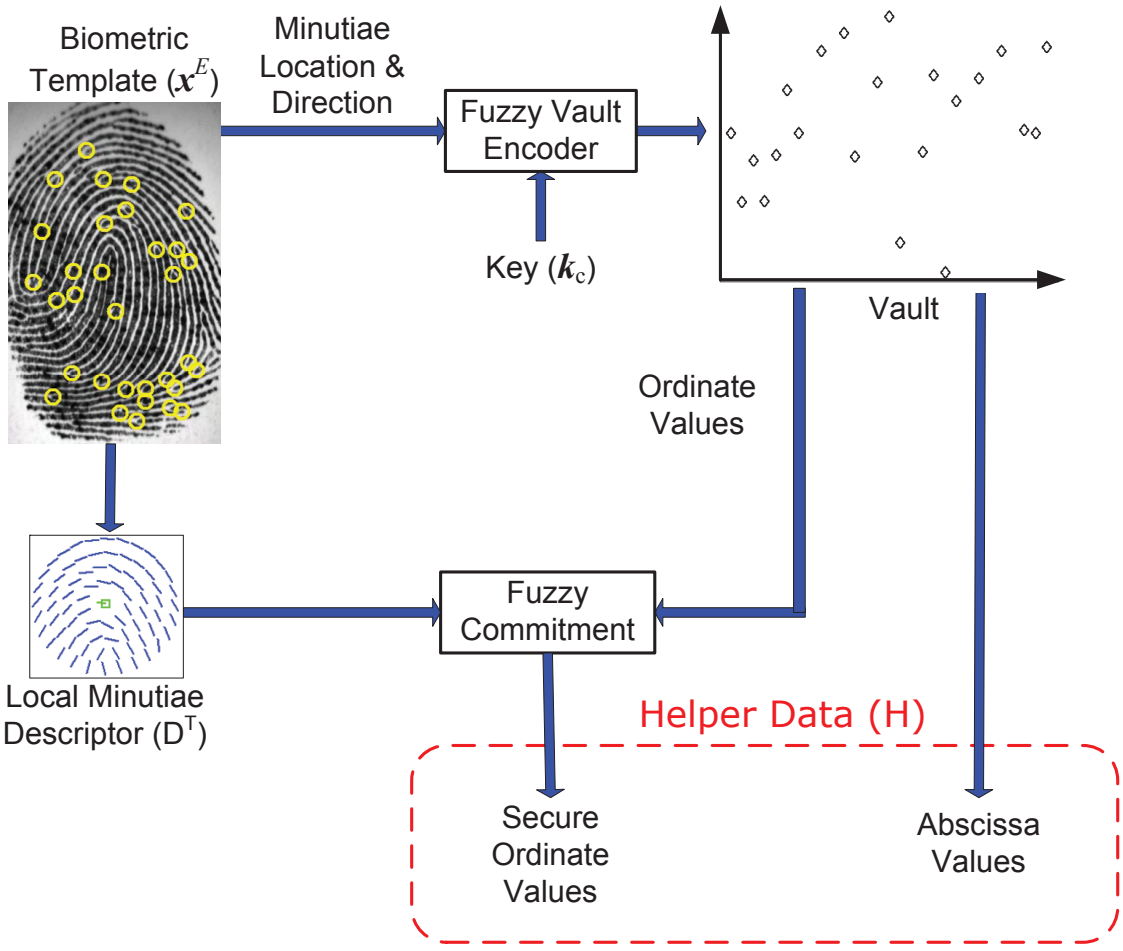


Figure 5.3: Fingerprint fuzzy vault encoding with minutiae descriptors.

values G and the high curvature points together constitute the proposed augmented fingerprint fuzzy vault.

During authentication (see Figure 5.4), the query fingerprint is first aligned using the high curvature points extracted from the template and query fingerprints as described in [91]. Then, r well separated and good quality minutiae are selected from the query and matched with the points in the vault in order to filter out most of the chaff points. Further, the minutiae descriptors are extracted from the query fingerprint and are binarized using the same procedure as in the enrolment stage. These binary descriptors are then used to recover the ordinate values from the associated fuzzy commitment. If the ordinate value is correctly decoded for some minimum number $(k+1)$ of genuine points in the vault, the degree k polynomial P is correctly reconstructed thereby indicating a successful match.

5.3.2 Descriptor Binarization

The fuzzy commitment scheme requires the biometric features to be in the form of a binary vector. Further, it is desirable that the Hamming distance among the matching and non-matching descriptors be as far apart as possible. In order to achieve this, we follow a four stage binarization scheme consisting of missing value estimation, dimensionality reduction, binarization and bit selection (see Figure 5.5).

Estimating Missing Values for Minutiae Descriptors

The descriptors corresponding to minutiae near the fingerprint boundary tend to have many missing values because only a part of the neighborhood of such minutiae lies within the fingerprint region (foreground). We estimate the missing values from the k-nearest descriptors of a given descriptor in the database that are expected to provide realistic and reliable estimates.

Since the orientation values have different characteristics than the ridge frequency

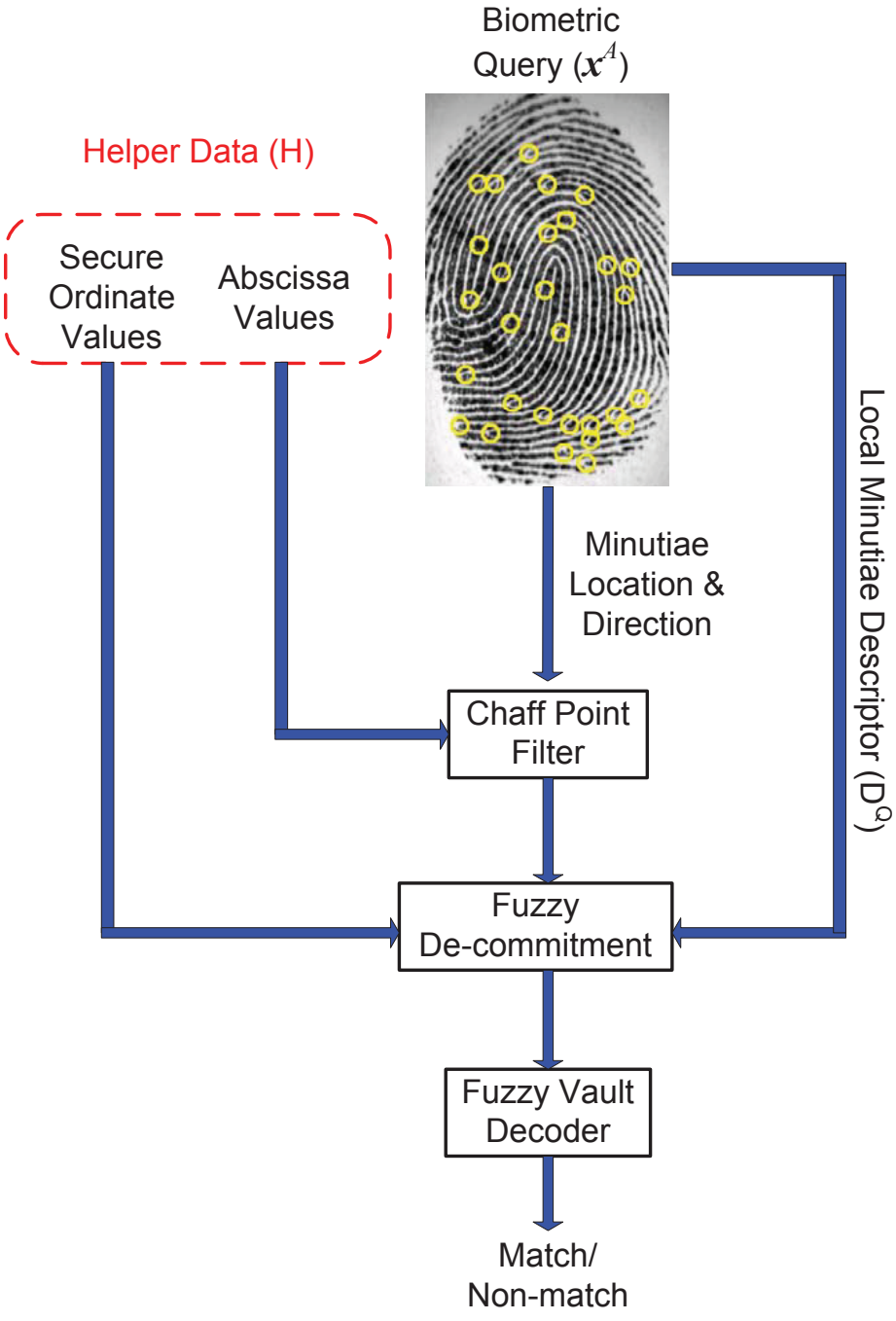


Figure 5.4: Authentication using the proposed fingerprint cryptosystem.

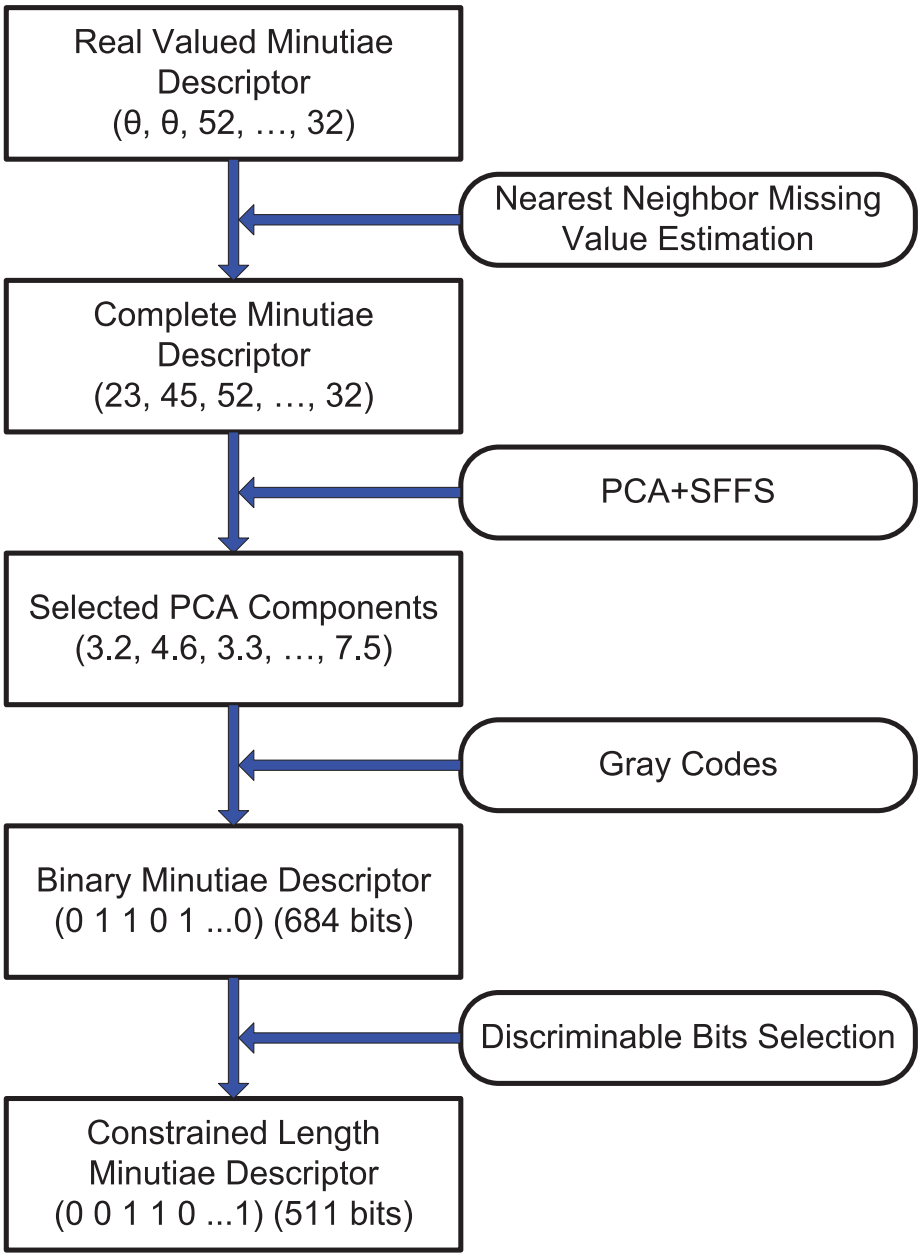


Figure 5.5: Different stages involved in obtaining a binary vector of desired length from raw minutiae descriptors.

values, missing values are estimated separately for these two types. We shall refer to the set of orientation values associated with the descriptor as the orientation descriptor and the set of ridge frequency values associated with the descriptor as the frequency descriptor.

The distance between two orientation descriptors $O^1 = \{o_1^1, o_2^1, ..., o_m^1\}$ and $O^2 = \{o_1^2, o_2^2, ..., o_m^2\}$ is given by

$$d(O^1, O^2) = \frac{\sum_{i=1}^{m} min(|o_i^1 - o_i^2|, 180 - |o_i^1 - o_i^2|) mask_{oi}}{\sum_{i=1}^{m} mask_{oi}} \tag{5.4}$$

where $mask_{oi}$ has a value 1 if both the o_i^1 and o_i^2 are inside the fingerprint region (foreground) and 0 otherwise. If the k nearest neighbors of ith orientation descriptor are $O^{(1)}, O^{(2)}, ..., O^{(k)}$ then the estimated orientation values are given by:

$$o_j^i = \frac{1}{2} atan \left(\frac{\sum_{l=1}^{k} \sin(2o_j^{(l)}) mask_{oj}^{(l)}}{\sum_{l=1}^{k} \cos(2o_j^{(l)}) mask_{oj}^{(l)}} \right) \tag{5.5}$$

where $mask_{oj}^{(l)}$ has value 1 if $o_j^{(l)}$ is in the foreground and 0, otherwise.

The missing values for the ridge frequency are also computed in a similar way by changing the distance measure between descriptors and the function that combines multiple descriptors to estimate the missing value. The distance between two frequency descriptors $F^1 = \{f_1^1, f_2^1, ..., f_m^1\}$ and $F^2 = \{f_1^2, f_2^2, ..., f_m^2\}$ is given by

$$d(F^1, F^1) = \frac{\sum_{i=1}^{m} |f_i^1 - f_i^2| mask_{fi}}{\sum_{i=1}^{m} mask_{fi}} \tag{5.6}$$

where $mask_{fi}$ has a value 1 if both the f_i^1 and f_i^2 are inside the fingerprint region (foreground) and 0, otherwise. The frequency values estimated from the k neighbors

of the ith descriptor are given by

$$f_j^i = \frac{\sum_{l=1}^{k} f_j^{(l)} mask_{fj}^{(l)}}{\sum_{l=1}^{k} mask_{fj}^{(l)}}, \tag{5.7}$$

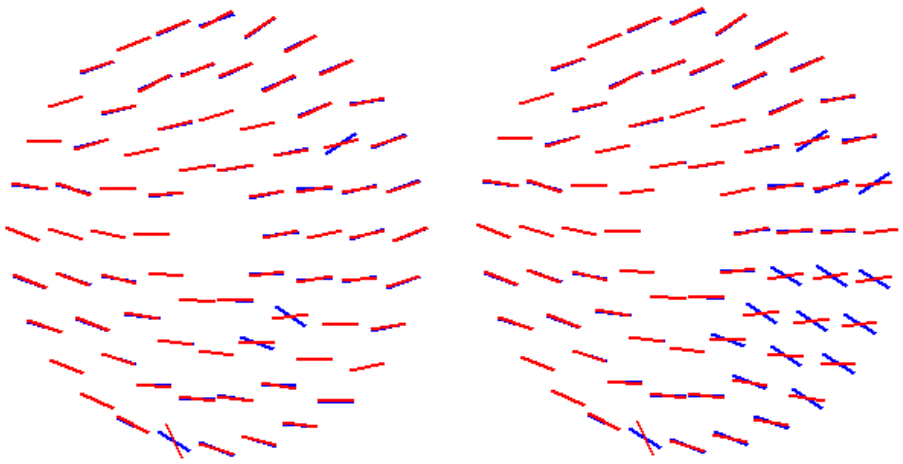
where $mask_{fj}^{(l)}$ has value 1 if $f_j^{(l)}$ is in foreground and 0, otherwise.

Note that we consider only those nearest neighbors where at least 75% of the values available in the given descriptor are also available in the selected neighbors as well. A small fraction of the descriptor values that could not be estimated using the above procedure were interpolated as a weighted average of the neighboring values. Figure 5.6 compares the orientation component of the descriptors where missing values were estimated using the nearest neighbor approach and the simple interpolation scheme. We observe that the values estimated using the nearest neighbor based technique is more similar to the real descriptor values in a matching descriptor (obtained from the same minutiae in a different impression of the same finger) compared to the simple interpolation scheme.

Dimensionality Reduction for Minutiae Descriptors

During fuzzy commitment decoding, if the difference between the enrolment and query biometric is such that the word to be decoded is farther than the error tolerance from any of the codewords of the error correcting code used, a decoding failure is detected. This happens very frequently in case the dimension of the code is large. This fact is detrimental to the security of the proposed scheme as it allows the attacker to decode each ordinate value separately. Note that instead of a decoding failure if an incorrect codeword is recovered, the attacker will not be aware of this until he tries to decode the complete fuzzy vault.

In order to avoid frequent decoding failures, we reduce the dimensionality of the minutiae descriptor using principal component analysis (PCA) [36] and sequential



(a) (b)

Figure 5.6: Estimating missing values in descriptors: (a) orientations of two matching descriptors overlaid where missing values were estimated using the nearest neighbor approach; (b) orientations of the same descriptors when simple interpolation is used for estimating the missing values. It can be observed that there are very few inconsistent orientation values in case the nearest neighbor approach is used.

forward floating search [103]. One of the motivations for using PCA is the fact that the different elements of minutiae descriptors are highly correlated, resulting in strongly correlated bits in the binarized descriptor. The use of PCA is expected to lead to uncorrelated bits in the binarized descriptor.

Since the orientation values do not belong to Euclidean space, a direct application of PCA is not expected to produce meaningful components. Thus, a new orientation descriptor is computed as:

$$O^{'} = [\cos(2o_1)\sin(2o_1)\cos(2o_2)\sin(2o_2)\cdots\cos(2o_m)\sin(2o_m)]\,. \tag{5.8}$$

The complete descriptor can be defined as

$$D = [\cos(2o_1)\sin(2o_1)\cos(2o_2)\sin(2o_2)..\cos(2o_m)\sin(2o_m)f_1f_2..f_m]\,. \tag{5.9}$$

PCA is now applied to the descriptor represented in Eq. 5.9 to obtain the uncor-

related components. Further, since certain components might be very noisy, we apply a supervised feature selection to select a subset of salient features. Among the various feature selection algorithms available [63,98], sequential forward floating selection algorithm (SFFS) [103] is simple to implement and provides good performance. We use the False Accept Rate (FAR) at the 98% Genuine Accept Rate (GAR) as the objective function to minimize for selecting salient features using SFFS. We use the Euclidean distance as the distance measure between the descriptors. Once the desired number of features is selected, they are binarized using the scheme described in Section 5.3.2.

Binarizing Minutiae Descriptors

In order to binarize the descriptors, we uniformly quantize the descriptor values into 2^{α} bins and use Gray code [52] to associate a binary string to each bin. Gray code has a property that the bit string associated with every codeword differs from that associated with its adjacent codeword by only one bit. Table 5.2 shows a 3 bit Gray code.

The discriminability index of each bit is defined as

$$\Gamma = \alpha_d \sigma_I - (1 - \alpha_d)\sigma_G, \tag{5.10}$$

Quantum index	Gray code
1	000
2	001
3	011
4	010
5	110
6	111
7	101
8	100

Table 5.2: 3-bit Gray code. Note that adjacent quanta differ in only a single bit.

where σ_G and σ_I are the intra-class and inter class variation of the ith bit, and $\alpha_d \in [0, 1]$ is a constant. Based on this discriminability index, a desired number of most discriminable bits are selected as the final bit string.

5.3.3 Experiments

We used the FVC2002 DB2 fingerprint database to compare the fuzzy vault performance with and without minutiae descriptors. Only the first two impressions of the 100 different fingers in the database were used in the experiments; one as the template and the other as the query. During both enrolment as well as authentication, the missing descriptor values are estimated using the 10-nearest neighbor approach as described in Section 5.3.2. The nearest neighbors are found among the descriptors corresponding to all the minutiae extracted from all images in FVC02 DB2; there are around $27,000$ descriptors in total. The orientation and frequency values of the descriptors are quantized separately into 2^5 and 2^4 values, respectively, and binarized using Gray codes as described in Section 5.3.2 to obtain 684 bits. From these 684 bits, 511 bits are selected using the bit selection scheme as described in Section 5.3.2. The BCH(511,19) [18] error correcting scheme is used for generating the fuzzy commitment that can correct upto 119 errors. Figure 5.7 shows the GAR and FAR values corresponding to the basic fuzzy vault implementation (without descriptors) and the proposed implementation where minutiae descriptors are used (Desc (511,19)). Failure to capture rate in both cases is 2%. We observe that the use of minutiae descriptors reduces the FAR of the system significantly, while the GAR remains nearly the same. For instance, when the degree of the polynomial is 6, the GAR is 95% for both the scenarios. However, the FAR is 0.7% when the descriptors are not used and 0.01% when the proposed cryptosystem is used. These estimates of GAR and FAR are based on 100 genuine matches and $9,900$ impostor matches.

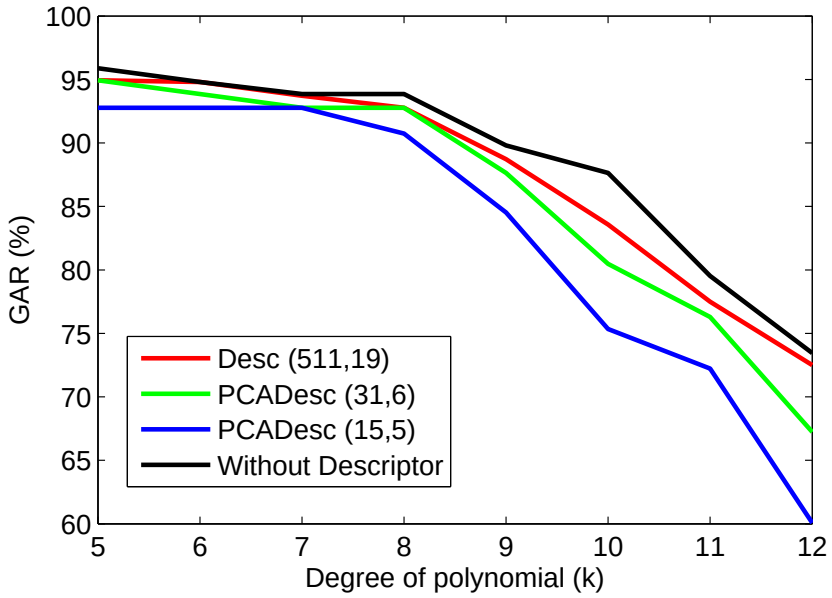
The principal component analysis (PCA) is further used to reduce the dimension-

ality of the descriptors as described in Section 5.3.2. The covariance matrix of the descriptors values, that is required for computing the principal components, is computed using the descriptors available in the database. First 10 principal components are retained and each one is quantized into 2^7 bins. A 7-bit Gray code is used to binarize each of the 10 components. Note that PCA and the desired components can be computed off-line once for all. Figure 5.7 shows the FAR as well as the GAR corresponding to 31-bit as well as 15-bit descriptors obtained by selecting 31 and 15 bits, respectively, from the available 70 bits. It can be seen that there is slight degradation in the GAR because of dimensionality reduction from 95% to 94% and 93%, respectively, for 31 and 15 bits descriptors. However, as described in Section 5.3.4, the security is increased by around 10 bits in case a 15-bit descriptor is used.

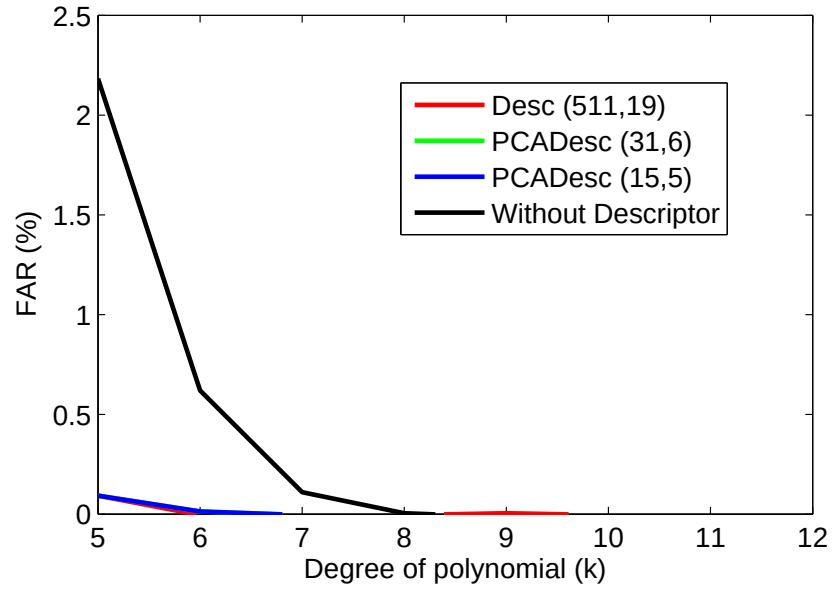
5.3.4 Security Analysis

In the proposed fingerprint fuzzy vault the true ordinate values can be obtained in two ways: (i) directly guessing the 16-bit ordinate values, and (ii) guessing the descriptors associated with each minutia. Since the ordinate values of the genuine points are obtained through an evaluation of a randomly generated secure polynomial, it is reasonable to assume that the difficulty of directly guessing an ordinate value is approximately 16 bits (assuming there are more than 16 information bits in the error correcting code, otherwise it is the number of information bits of the code). Also since the adversary has to simultaneously guess $(k+1)$ ordinate values correctly, this corresponds to approximately $16(k+1)$ bits of security.

In order to estimate security against guessing the descriptor, let the entropy of a minutia descriptor D be I_D bits and say ρ bits out of these should be corrected. As shown by Hao et al. [54], the difficulty in guessing a minutiae descriptor is approximately $R = \log\left(2^{I_D}/\binom{I_D}{\lceil\rho\rceil}\right)$ bits. Since the adversary has to simultaneously guess $(k+1)$ minutiae descriptors correctly, using minutiae descriptors provides approxi-



(a)



(b)

Figure 5.7: GAR (a) and FAR (b) for the fuzzy vault with and without descriptors. "Desc (511, 19)" corresponds to case when orientation values are quantized into 2^5 quanta, ridge frequency values are quantized into 2^4 quanta and 511 bits are extracted from them. Here the fuzzy commitment scheme is constructed using BCH(511,19) code. "PCADesc (31,6)" and "PCADesc (15,5)" correspond to cases when 10 principal components are extracted and each value is divided into 2^7 quanta. In "PCADesc (31,6)", 31 bits are extracted and BCH(31,6) code is used for fuzzy commitment whereas in "PCADesc (15,5)" 15 bits are extracted and BCH(15,5) code is used. BCH(511,19) corrects up to 119 errors, BCH(31,6) corrects up to 7 errors and BCH(15,5) corrects up to 3 errors.

mately $(k+1)R$ bits of security. Although the length of descriptor is N bits, there is a strong correlation between the descriptor bits leading to a reduction in effective entropy of the descriptor bits i.e. I_D. We empirically determine that approximately $N/4$ bit errors need to be corrected in order to preserve the GAR to a large extent. Thus ρ can be approximated as $I_D/4$. Thus, if $k = 8$ and $I_D = 6$ bits, then $R \approx 2$ bits. In this scenario, the proposed scheme increases the security of the fuzzy vault by approximately 18 bits, so the overall security now becomes 49 $(31 + 18)$ bits. This is equivalent to a 6 character password.

The above security analysis assumes the use of a *perfect* error correction coding scheme (a w-error correcting binary code of size 2^N is said to be perfect if for every word $C^{'}$, there is a unique codeword C such that the Hamming distance between C and $C^{'}$ is at most w bits). It has, however, been proven that any non-trivial perfect code over a prime-power alphabet has the parameters of a Hamming code or a Golay code [58]. Note that Hamming codes correct only single errors whereas Golay codes correct only up to three errors in code of length 24 bits and thus they would not be applicable to the current problem.

If the coding scheme is not perfect, some of the words may result in a decoding failure which would indicate an incorrect minutia descriptor being used to de-commit the ordinate value. Due to the unknown distribution of biometric features, it is important to empirically estimate the number of decoding failure and incorrect decodings while using a particular error correcting code. Note that even if all the incorrect descriptors lead to decoding failure, the security is at least as good as the security of the original fuzzy vault.

When an adversary applies a descriptor to decode the secure ordinate value, following situations can arise: i) a decoding failure is detected, ii) the correct codeword c is obtained, or iii) an incorrect codeword $c_i (\neq c)$ is obtained.

We are interested in estimating the relative frequency of these three events as

they provide an estimate of ambiguity about the true codeword. Let the relative frequency of the three events be: π_{df}, π_0, and $\pi_i, i = 1, 2, ...$ in order. One strategy an adversary might employ would be to try decoding the ordinate value with a large number of descriptors one by one and select the first ordinate value decoded. Here the adversary would be successful with probability

$$p_a = \left\{ \frac{\pi_0}{\sum_i \pi_i} \right\}^{(k+1)} = p_0^{(k+1)}, \tag{5.11}$$

where $p_0 = \frac{\pi_0}{\sum_i \pi_i}$. Thus the number of bits of security added would be equal to $T_a = -\log_2 p_a$.

In order to estimate π_{df}, π_0, and π_i, we randomly selected 20 different descriptors and tried to decode those using the database containing $27,000$ descriptors. Table 5.3 shows the values corresponding to π_{df}, π_0, $max_i \pi_i$ and T_a for the different representations of descriptors considered. It can be seen that $BCH(31, 6)$ provides around 7 bits of security, on average, whereas that $BCH(15, 5)$ provides around 28 bits.

In our experiments with the imperfect codes having high dimension e.g. BCH (511,19) or BCH(31,5), it has been observed that $\pi_0 >> \pi_i$. This can be explained by the fact that if the difference between two matching descriptors is less than the error correction capacity of the code, which is often the case, the errors introduced in the codeword still leads to correct decoding. On the other hand, when a randomly selected descriptor is used to decode the fuzzy commitment, a large number of errors beyond the error correcting capacity is introduced into the codeword. Due to this, the codeword is shifted to a non-decodable region with high probability leading to a decoding failure.

Note that in case of BCH(511,19), theoretical estimate of the fraction of space that is not decodable is approximately $1 - 10^{-19}$, that for BCH(31,6) is approximately 0.9 and for BCH(15,5), it is approximately 0.44 which is consistent with the probabilities

Descriptor Format	π_{df}		
	min	max	median
Desc (511,19)	0.941	0.999	0.991
PCADesc (31,6)	0.761	0.896	0.826
PCADesc (15,5)	0.394	0.448	0.418
	π_0		
Desc (511,19)	0.000	0.059	0.001
PCADesc (31,6)	0.032	0.194	0.103
PCADesc (15,5)	0.014	0.168	0.070
	$max_i(\pi_i)$		
Desc (511,19)	0	0	0
PCADesc (31,6)	0.007	0.050	0.016
PCADesc (15,5)	0.048	0.128	0.074
	T_a		
Desc (511,19)	0	0	0
PCADesc (31,6)	2.74	17.09	6.63
PCADesc (15,5)	16.61	48.41	27.55

Table 5.3: The values corresponding to π_{df}, π_0, $\max_i(\pi_i)$ and T_a for the different representations of descriptors considered.

of decoding failure reported in Table 5.3. Also, no incorrect decoding was detected in case of using BCH(511,19) due to a large fraction of non-decodable region.

Another strategy that an adversary can employ is to use t different descriptors for decoding each secure ordinate value and get the ordinate value that repeated the maximum number of times. Note that, on average, there would be $u = t(1 - \pi_{df})$ different descriptors that will not produce decoding failures. Thus the adversary will succeed if there are more than up_i^{max} correctly decoded ordinate values among the set of u decoded values, where

$$p_i^{max} = \left\{ \frac{\max_i\{\pi_i; i = 1, 2, ...\}}{\sum_i \pi_i} \right\}. \tag{5.12}$$

Thus the probability of successful attack is given by

$$p'_a = \left\{ p\left(\#(\text{correct codewords}) > \lceil up_i^{max} \rceil\right) \right\}^{(k+1)}, \tag{5.13}$$

where

$$p\left(\#(\text{correct codewords}) > l\right) = \sum_{i=l+1}^{u} \binom{u}{l} p_0^i (1-p_0)^{u-i}. \tag{5.14}$$

Note that the security in terms of number of bits is given by $T_a' = -\log_2 p_a'$.

We assessed the variation in the number of bits of security as u increases corresponding to the case when the descriptor is represented as a 15-bit vector. It is noted that in more than half of the cases, around 10 bits of security can be imparted to the fuzzy vault in case the degree of polynomial secured by the fuzzy vault, i.e. k, is 8.

The increase in the number of descriptors tried by the adversary, i.e. t, also leads to increased computational requirement. Thus even though no additional information theoretic security is imparted in case of "BCH(511,19)", there is significant computational cost to the adversary in order to compromise the system due to large π_{df} leading to improvement in security to a certain extent. Note that given u, t is directly proportional to π_{df}.

5.4 Summary

In this chapter, we have discussed two ways to modify and improve a fingerprint fuzzy vault. In the first modification, we incorporate a user password in the fuzzy vault which leads to a significant increase in security as well as matching performance by increasing the randomness in minutiae location and direction. Moreover, in case the template is compromised, a new template can be constructed using a different password. The second improvement proposed in this chapter allows incorporating information in the neighborhood of minutiae in the fuzzy vault. Thus, even if the attacker is able to find the genuine minutiae in the vault, he will still not be able to decode the vault unless he has knowledge about the minutiae neighborhood.

Chapter 6

Template Transformation

6.1 Introduction

Template transformation techniques constitute the second major category of the software based biometric template protection techniques besides biometric cryptosystems. In a template transformation technique, the biometric template is transformed based on parameters derived from a user's password or a key and the transformed template is stored in the system during enrolment. During authentication, the query biometric is similarly transformed and is matched with the transformed template for an accept/reject decision. A major advantage of such techniques is that the original biometric is never revealed in the system. Furthermore, it ensures non-linkability as it is usually difficult to determine if two transformed templates are obtained from the same biometric (e.g., same finger) or not. Techniques have also been designed to ensure non-invertibility even if the transformation key is available. That is, even if the parameters of the transformation are available, it is difficult to recover the original template given the transformed template.

One of the major criticisms of template protection techniques is that their security has not been thoroughly analyzed [64]. In this chapter, we provide a comprehensive

set of metrics to estimate the vulnerability of a template transformation technique against various vulnerabilities mentioned in Section 1.3. We specifically analyze the security of two well known template transformation techniques, namely, biohashing and cancelable fingerprint templates based on the proposed metrics. Our analysis indicates that both these schemes are vulnerable to intrusion and linkage attacks because it is relatively easy to obtain either a close approximation of the original template, as in the case of biohashing approach, or a pre-image of the transformed template as in the case of cancelable fingerprints approach.

6.2 Background

A number of template transformation techniques have been proposed in literature (see Table 6.1). These techniques can be classified into two main categories based on the specific representation of the biometric template. These categories are: (i) vector based transformation techniques and (ii) interest points based transformation techniques.

6.2.1 Vector based transformation techniques

In the case of vector based techniques, the biometric templates are represented as a real or binary vector and the dissimilarity between two vectors is usually computed using the Euclidean distance measure. One of the main requirements of a vector based template transformation function is the preservation of distances between the vectors after transformation. Biohashing [122] is one such technique (see Figure 6.1), where the feature vector is transformed by multiplying it with a user specific orthogonal transformation matrix and thresholding the individual elements. Due to increased inter-class variation and preservation of intra-class variation, biohashing significantly improves the matching performance. However, if the key, and thus the user specific or-

Table 6.1: List of different template transformation techniques available in literature and their characteristics.

Technique	**Trait**	**Features**	**Transformation**	**Final representation**
Biohashing [122, 123], PalmHash [31]	Face, Palmprint, Fingerprint	Vector (Fisher Discriminant Features)	Random matrix multiplication	Vector
BioPhasor [124]	Fingerprint	Vector (FingerCode)	Non-linear	Vector
Cancelable Face [111]	Face	Vector (Face image)	Random matrix convolution	Vector
Robust Hash [121]	Face	Vector (Singular values of face image matrix)	Smooth multimodal function evaluation	Vector
Class Distribution Preserving (CDP) Transformation [46]	Face	Vector (Fisherface features)	Evaluation of distance of the feature vector from a set of points	Vector
Cancelable Iris [142]	Iris	Vector (Log-Gabor response)	Circular shift and combination, adding new pattern	Vector
Histogram of minutiae triangles [38]	Fingerprint	Interest point	Hashing the histogram of minutiae triangle features	Vector
Symmetric Hash [127]	Fingerprint	Interest point (Minutiae as complex numbers)	Set of order invariant functions of minutiae	Minutiae map
Cancelable Fingerprints [106]	Fingerprint	Interest point (Minutiae map)	Image folding	Minuitae map
Alignment free cancelable fingerprint [73]	Fingerprint	Interest point (minutiae map, orientation field)	Transform minutiae according to surrounding orientation field	Minutiae map
Cuboid based Minutiae Aggregates [120]	Fingerprint	Interest point (Minutiae map)	Minutiae aggregate feature selection from random local regions	Vector

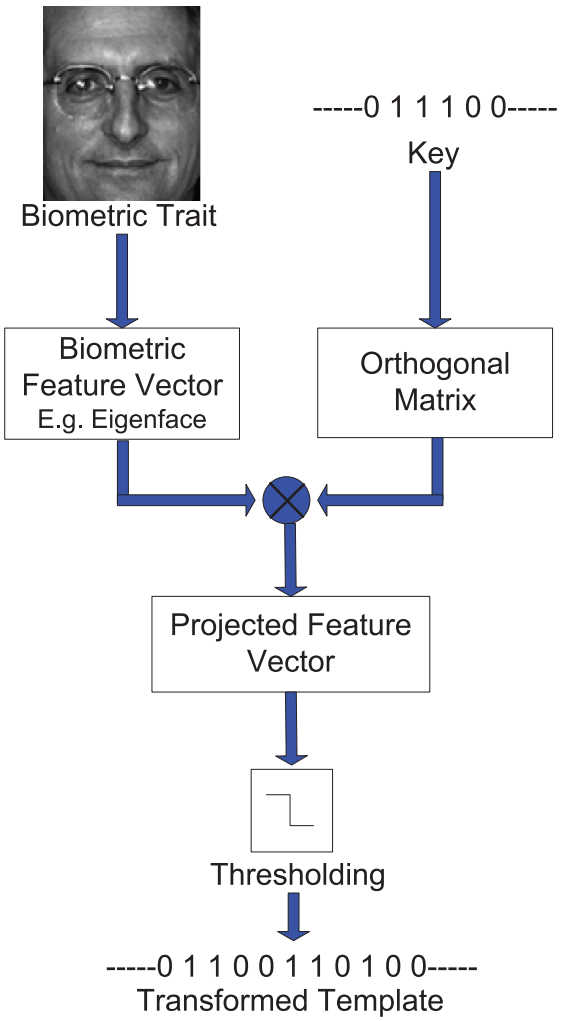


Figure 6.1: Schematic diagram of the biohashing technique.

thogonal transformation matrix, is known to the adversary, the matching performance typically degrades due to the quantization of features and dimensionality reduction. Another drawback of the biohashing scheme is that it is easy to invert and recover the original biometric feature vector when the key is known to the adversary (see Section 6.5). Note that in the context of template transformation techniques, invertibility is measured in terms of the computational complexity and the number of guesses involved in recovering the original template. The complexity of recovering the biometric sample from the recovered original biometric template is not included in order to focus the analysis on the template transformation technique and not on the tem-

plate representation itself. In some cases such as biohashing, it is straightforward to directly recover the original biometric template (or a close approximation of it) when the key is known. However, in other cases like robust hashing [121] and cancelable fingerprint templates [106], it is either computationally hard to obtain the complete pre-image[1] of the transformed template or difficult to identify the original biometric template from the pre-image due to the large size of the pre-image. Such schemes are considered to be difficult to invert (also loosely referred to as "non-invertible").

An improvement of the biohashing scheme is the biophasor [124] technique, where the rows of the orthogonal transformation matrix are used as the imaginary part and added to the biometric vector to obtain a set of complex vectors. For each of these vectors, the argument of the complex values in them are averaged and quantized to form the final binary template. This transformation has been shown to better preserve the matching performance even if the password is known to the adversary. Although this scheme is claimed to be non-invertible, the complexity involved in inverting this transformation is not known. Savvides et al. [111] showed that the distance between two Minimum Average Correlation Energy (MACE) filter outputs is preserved even when the face image is convolved using a random kernel matrix for template protection. However, this scheme is invertible given the knowledge of the convolution kernel and the specific MACE filters used. Sutcu et al. [121] proposed a transformation technique, where each element of the input biometric vector is evaluated on a multi-modal polynomial. Due to the many-to-one nature of the transformation function induced by the multi-modality of the polynomials, it is difficult to invert the transformed template. Feng and Yuen [45] transformed the template by randomly selecting a set of vectors of the same dimension as the biometric feature vector and then storing the Euclidean distances of the biometric vector from these vectors. This technique

[1] A pre-image of a transformed template is the collection of all the templates in the original domain that can generate the given transformed template.

assumes knowledge of the feature distribution of individual users while designing the transform, which possibly leaks some additional information regarding the biometric vector. The complexity of inverting the template i.e. recovering the original biometric from the transformed template is expected to be greater than that of biohashing technique. Zuo et al. [142] proposed two template transformation schemes for iris images. In the first scheme called "Combo", the original iris template was tessellated into rectangles, rows were cyclically shifted and different rows were added to obtain the transformed template. In the second scheme called "Salting", the iris image or its binary representation was added to a randomly generated texture to obtain the transformed template. The "Combo" approach is shown to be difficult to invert because of the addition of two different biometric features, which provides ambiguity about the component features.

6.2.2 Interest point based template transformation

Fingerprints are most commonly represented by a set of points, called minutiae. Hence, many fingerprint template transformation techniques are based on minutiae as the initial representation. Furthermore, to use the available minutiae-based fingerprint matchers in the transformed domain, it is desirable to have the final representation also in the form of a set of minutiae. To satisfy this criterion, Ratha et al. [106] proposed the use of cancelable fingerprint templates designed using three different minutiae transformation techniques, namely, cartesian, polar and functional (see Figure 6.2). Note that the term cancelable means that the template can be canceled or revoked if it is exposed to an adversary and an adversary having the stolen template will not be able to gain any significant knowledge about the replaced template. Cancelability thus directly implies non-linkability and vice versa. In the cartesian transformation, the fingerprint is regularly tessellated into a set of rectangles and these rectangles are displaced according to the associated key. The polar

transformation is similar to the cartesian transformation except that the fingerprint is divided into a number of concentric shells and each shell is divided into sectors. Since the size of sectors is different for different shells, some restrictions are placed on the displacement of the sectors based on the password. In case of the functional transformation, two different functions are used: a mixture of 2D Gaussians and electric potential field in 2D charge distribution. These functions are evaluated at the minutiae locations to obtain the translation corresponding to that minutia.

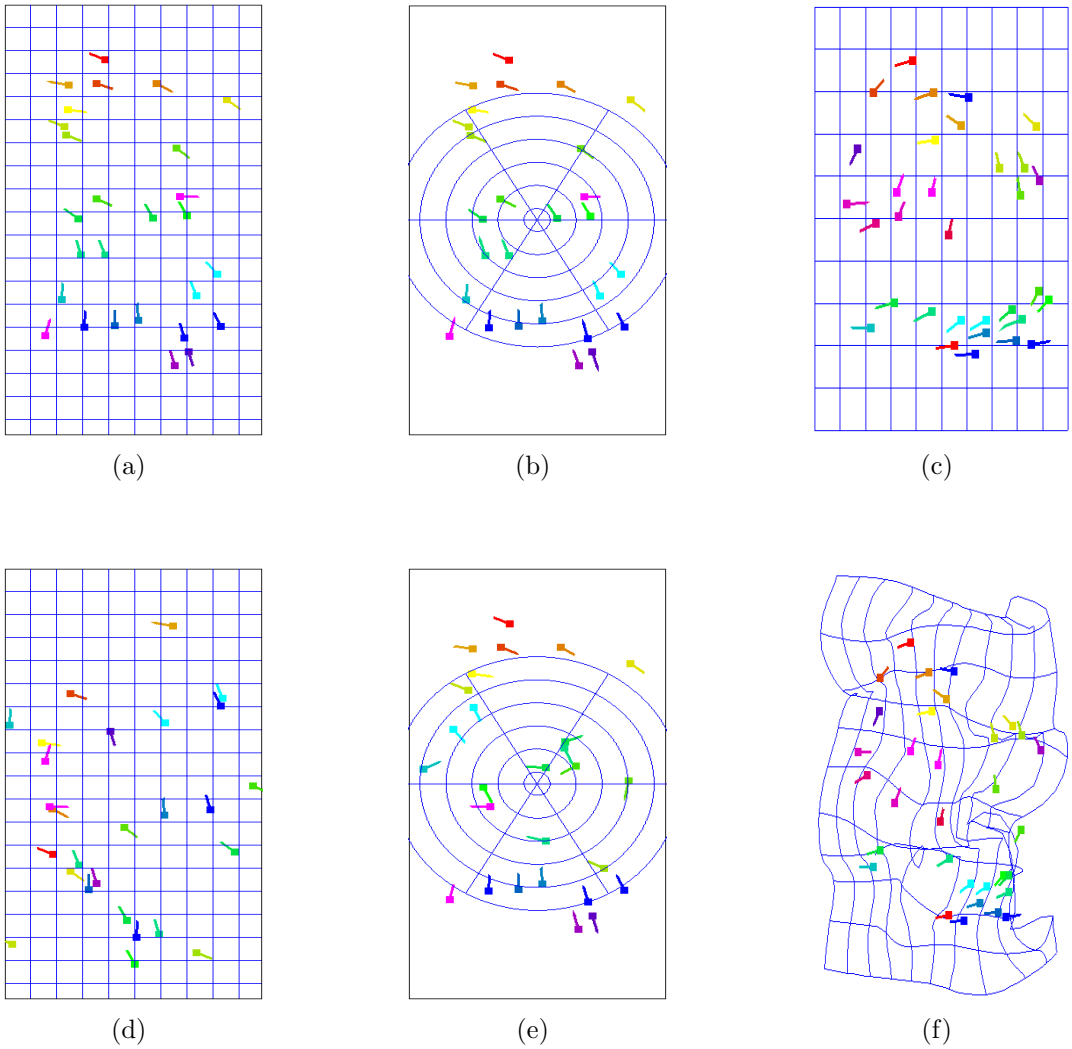


Figure 6.2: The original and transformed fingerprints for (a,d) Cartesian, (b,e) polar, and (c,f) Gaussian mixtures based transform .

All the three transformations proposed by Ratha et al. [106] are claimed by them to be difficult to invert due to the many-to-one nature of the transformation functions. However, these techniques lead to a reduction in the matching performance due to an increase in the intra-user variations[2]. Such transforms also require the fingerprints to be accurately aligned before applying the transformation as misalignment can further increase intra-user variations. To avoid alignment, Lee et al. [73] proposed an alignment-free cancelable fingerprint transform. In this scheme, each minutia is transformed according to the orientation field around that minutia, which makes the relative translation of the minutia invariant to the positioning of the finger. Tulyakov et al. [127] use each minutia along with its two nearest neighbors to select one of the several so called symmetric functions. The selected symmetric function is then evaluated on the three minutiae to obtain the coordinates of the transformed minutia.

Techniques have also been proposed to convert the minutiae based representation into a vector based representation. Farooq et al. [38] select all minutiae triplets satisfying certain criteria and construct a histogram. Only those bins in the histogram with a single element are retained and the remaining bins are emptied to obtain the final binary feature vector. Cancelability is further induced in this representation by flipping some of the bits and permuting the binary vector based on a specific key. The limitation of this approach is that it is easy to determine the unique triangles present in the fingerprint and the sides with similar length can be matched and combined to construct an approximate minutiae distribution. The complexity of such a procedure however might be high. Another scheme proposed by Sutcu et al. [120], converts a set of minutiae into a vector based representation by counting the number of minutiae falling in certain specified rectangular regions. The configurations of rectangular

[2]Intra user variation refers to changes in the template of the same user in different acquisitions of the biometric sample. Since the transformation functions are generally non-Euclidean, variations in minutiae position and orientation are escalated due to transformation, leading to high false reject rate.

regions can be changed in order to generate another template from the same biometric thereby inducing cancelability.

6.3 Analysis of Template Transformation

We analyze the performance of a template transformation technique in terms of its usability, security against intrusion and linkage threats. See Chapter 1 for a discussion on template intrusion and linkage. We employ the following notation to describe the security metrics.

- $\mathbf{x}_z^E$ and $\mathbf{x}_z^A$: The template and query biometric features of user z, respectively.
- f: The feature transformation function; f^{-1} denotes its inverse.
- f_β^{-1}: The partial inverse transformation function, where β is the fraction of the original biometric template obtained by inverting the transformed template.
- K_z: A set of transformation parameters corresponding to user z; $K_z^{'}$ is a different set of transformation parameters for the same user.
- $\mathcal{D}_O$: A distance function between the biometric features in the untransformed (original) domain; $\mathcal{D}_T$ is a distance function between the biometric features in the transformed domain.
- ϵ: The system threshold such that the biometric system declares a "match" if the distance between the template and query biometric features is less than a threshold ϵ.

6.3.1 Evaluation Measure for System Usability

Security of a biometric recognition system affects the usability of the system as well. While considering the system security, it is important to measure any inconvenience

incurred to the genuine users of the system as a result of the security techniques implemented. We measure the usability in terms of the false reject rate (FRR) of the system. Note that it would be equally appropriate to measure the system usability using false non-match rate (FNMR) or genuine accept rate (GAR). The false reject rate of the biometric system prior to the template transformation, FRR_O, is given by

$$FRR_O(\epsilon) = P\left(\mathcal{D}_O\left(\mathbf{x}_z^E, \mathbf{x}_z^A\right) \geq \epsilon\right). \tag{6.1}$$

The false reject rate of the biometric system after the application of template transformation, FRR_T, is

$$FRR_T(\epsilon) = P\left(\mathcal{D}_T\left(f\left(\mathbf{x}_z^E, K_z\right), f\left(\mathbf{x}_z^A, K_z\right)\right) \geq \epsilon\right). \tag{6.2}$$

FRR_O and FRR_T depend on the system threshold ϵ and must be as low as possible to avoid inconvenience to the users. The threshold ϵ also controls the security and privacy of the system because the probability of success of an intrusion or linkage attack depends on ϵ.

6.3.2 Security Evaluation Measures for Intrusion Threats

First, we consider the zero effort attack where an impostor presents his own biometric trait in order to get authenticated. The intrusion success probability for this attack is measured in terms of the false accept rate. The false accept rate of the biometric system when no template transformation is performed is given by

$$FAR_O(\epsilon) = P\left(\mathcal{D}_O\left(\mathbf{x}_i^E, \mathbf{x}_j^A\right) < \epsilon\right), \text{where } i \neq j. \tag{6.3}$$

A plot of FAR_O versus $(1 - FRR_O)$ for various values of ϵ gives the receiver operating

characteristic (ROC_{orig}) curve of the biometric system prior to template transformation.

Given a biometric system with stored transformed templates, the impostor has to present the biometric features along with a set of transformation parameters for authentication. This entails two scenarios: one where the impostor knows the transformation parameters and second where he does not know the transformation parameters. Suppose that the impostor does not know the transformation parameters of the specific user he is trying to impersonate. The FAR with unknown transformation parameters (K) is given by

$$FAR_{UK}(\epsilon) = P\left(\mathcal{D}_T\left(f\left(\mathbf{x}_i^E, K_i\right), f\left(\mathbf{x}_j^A, K_j\right)\right) < \epsilon\right), \text{where } i \neq j. \tag{6.4}$$

and a plot of FAR_{UK} versus $(1 - FRR_T)$ gives the corresponding receiver operating characteristic (ROC_{diff}) curve of the biometric system.

If the impostor somehow knows the transformation parameters of the genuine user that he is trying to impersonate, the FAR with known transformation parameters (K) is

$$FAR_{KK}(\epsilon) = P\left(\mathcal{D}_T\left(f\left(\mathbf{x}_i^E, K_i\right), f\left(\mathbf{x}_j^A, K_i\right)\right) < \epsilon\right), \text{where } i \neq j \tag{6.5}$$

and a plot of FAR_{KK} versus $(1 - FRR_T)$ gives the corresponding receiver operating characteristic (ROC_{same}) curve. A comparison of ROC_{orig} and ROC_{same} will indicate the degradation in the matching performance due to the template transformation.

Besides the false accept rates, two other intrusion probabilities must be considered. First we consider the case when the stored (transformed) template and the

transformation parameters are available to the adversary. In this case, the adversary tries to recover either a fraction (β) or the complete biometric template and then replay the inverted template along with the transformation parameters to gain access fraudulently. The probability of success of such an attack is called the Intrusion Rate due to Inversion for the Same biometric system ($IRIS$) and is defined as

$$IRIS(\beta, \epsilon) = P\left(\mathcal{D}_T\left(f\left(f_\beta^{-1}\left(f\left(\mathbf{x}_i^E, K_i\right), K_i\right), K_i\right), f\left(\mathbf{x}_i^E, K_i\right)\right) < \epsilon\right). \quad (6.6)$$

Note that this measure is similar to the genuine accept rate corresponding to the same impression scenario discussed in Chapter 2. The MCC-B representation discussed in Chapter 2 is, however, not a template transformation technique since no key is involved in generating MCC-B from a fingerprint image and thus multiple MCC-B templates generated from the same finger are easily linkable. The value of $IRIS(\beta, \epsilon)$ is usually 1 if a transformation is easy to invert or an element in the pre-image of the transformed template can be obtained (as in the case of many-to-one transformations). $IRIS(\beta, \epsilon)$ will be low when it is difficult to obtain any element in the pre-image of the transformed template.

Next, we consider the case when the stored (transformed) template and the transformation parameters are available to the adversary who wants to impersonate the same user in a different biometric system that employs the same biometric trait. We also assume that the adversary has knowledge of the transformation parameters of the second system. In this case, the adversary will try to recover either a fraction (β) or the complete biometric template and then replay the inverted template along with the transformation parameters of the second system to gain access fraudulently. The probability of success of such an attack is referred to as the Intrusion Rate due to Inversion for a Different biometric system ($IRID$) and is defined as

$$IRID(\beta,\epsilon) = P\left(\mathcal{D}_T\left(f\left(f_\beta^{-1}\left(f\left(\mathbf{x}_i^E, K_i\right), K_i\right), K_i'\right), f\left(\mathbf{x}_i^A, K_i'\right)\right) < \epsilon\right). \quad (6.7)$$

This measure is similar to the accept rate corresponding to the different impression scenario discussed in Chapter 2.

Finally, we also need to consider the effort spent by the adversary to invert a transformed template. Let $E(\beta)$ denote the effort required in terms of the number of guesses required (expressed in bits) to recover a fraction β of the original biometric template from the transformed template. The plot of β versus $E(\beta)$ is called the coverage-effort curve (C-E curve) [87]. The coverage-effort curve is a quantitative measure to evaluate the invertibility of a biometric template, provided it is possible for the adversary to check whether the recovered template is a true template. Also note that the evaluation of the fraction β is dependent on the kind of biometric template considered. The C-E curve relates the probability of success of intrusion attacks due to inversion ($IRIS$ and $IRID$) with the difficulty in inverting a transformed biometric template.

6.3.3 Security Evaluation Measures for Linkage Threats

In order to link two different templates generated from the same biometric trait of a user with different sets of transformation parameters, the adversary may either directly match the transformed templates or he can first invert the templates and then match the inverted templates. Suppose that both sets of transformation parameters, which were used to generate the two templates, are known to the adversary. The cross match rates can be defined in the transformed (CMR_T) and original (CMR_O) feature domains as follows.

$$CMR_T(\epsilon) = P\left(\mathcal{D}_T\left(f\left(\mathbf{x}_i^E, K_i\right), f\left(\mathbf{x}_i^A, K_i^{'}\right)\right) < \epsilon\right), \text{ and} \tag{6.8}$$

$$CMR_O(\beta, \epsilon) = P\left(\mathcal{D}_O\left(f_\beta^{-1}\left(f\left(\mathbf{x}_i^E, K_i\right), K_i\right), f_\beta^{-1}\left(f\left(\mathbf{x}_i^A, K_i^{'}\right), K_i^{'}\right)\right) < \epsilon\right). \tag{6.9}$$

In case of linkage attack in the original domain, the failure rate or the False Cross Match Rate of the attacker is given by

$$FCMR_O(\beta, \epsilon) = P\left(\mathcal{D}_O\left(f_\beta^{-1}\left(f\left(\mathbf{x}_i^E, K_i\right), K_i\right), f_\beta^{-1}\left(f\left(\mathbf{x}_j^A, K_j^{'}\right), K_j^{'}\right)\right) < \epsilon\right), \tag{6.10}$$

where $i \neq j$. A plot of $CMR_O(\beta, \epsilon)$ versus $FCMR_O(\beta, \epsilon)$ provides the receiver operating characteristic (ROC_{inv}) curve for the linkage attack in the original domain.

The complexity of cross-matching biophasors is difficult to estimate, however, inversion of biohashing, and cancelable face is computationally easy and is expected to generate a close approximation to the original template. In order to link templates secured using cancelable fingerprint templates approach, one can overlay all the pre-images of minutiae in the transformed template to obtain an aggregate template. It is expected that these aggregate template will have a large number of matching minutiae even if the templates are constructed by applying two different transformations to a fingerprint [104,116]. Note that in this case the matcher should not penalize the non-matching minutiae. Similar techniques can also be used to link templates encrypted using the robust hashing approach. In case of histogram of minutiae triplets, it is easy to obtain the original histogram, which can be easily matched. Symmetric hashing, cancelable iris, CDP, and cuboid based minutiae aggregates are not straight forward to invert and link.

A comprehensive security evaluation of a template transformation scheme entails analysis of the intrusion and linkage probabilities and their effect on the system usability measured in terms of FRR_T. In order to measure the probability of system intrusion, we have defined FAR_{UK}, FAR_{KK}, $IRIS$, and $IRID$. Note that FAR_{UK} and FAR_{KK} analyze the attacks staged by an adversary by presenting an arbitrary biometric template whereas $IRIS$ and $IRID$ analyze the attacks when the attacker steals a transformed template, inverts it and then uses it for intruding the system. Linkage probabilities can be measured in terms of CMR_O, where the templates are linked in the original domain (after inversion), and CMR_T, where the templates are linked in the transformed domain.

6.4 Security of Cancelable Fingerprint Templates

We choose cancelable fingerprint templates as an example for security evaluation because though the scheme is difficult to invert, a pre-image computation technique is available in the literature [87]. We evaluate the security strength of the mixture of Gaussians based transformation function, which is claimed to have the best performance among all the transforms evaluated by Ratha et al. [106]. The mixture of Gaussians used to obtain the transformation function is given by

$$f(\vec{x}) = \sum_{i=1}^{N} t_i \pi_i e^{-\frac{1}{2}(\vec{x}-\vec{\mu_i})\Sigma_i^{-1}(\vec{x}-\vec{\mu_i})'}, \tag{6.11}$$

where N is the number of mixture components, and π_i, t_i, μ_i, and Σ_i correspond to the mixing probabilities, the signs (+ or -), mean vectors, and covariance matrices of the different components, respectively. Here, $\vec{x}$ is a vector representation of a minutia point consisting of only the x and y coordinates of the minutiae. In our experiments, N is set to 24 and Σ_i is taken to be a diagonal matrix with each diagonal entry equal to 50^2 for each component. The remaining parameters are determined using the user

specific key. These parameters are the same as those used by Ratha et al. [106].

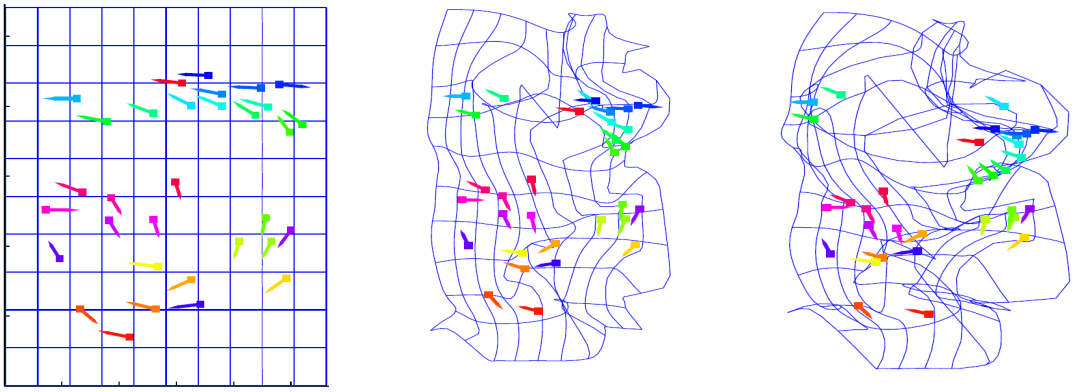
The transformation of each minutia is represented as direction of minutia translation (denoted by ϕ_ψ), magnitude of minutia translation (denoted by ϕ_d) and difference in minutia direction (denoted by ϕ_θ) . The three components of the transformation can be obtained as:

$$\begin{aligned} \phi_d(\vec{x}) &= \gamma\{1 + f(\vec{x})\} \\ \phi_\psi(\vec{x}) &= \arctan(\frac{g'_y(\vec{x})}{g'_x(\vec{x})}) + \alpha_\psi \\ \phi_\theta(\vec{x}) &= \arctan(\frac{f'_y(\vec{x})}{f'_x(\vec{x})}) + \alpha_\theta \end{aligned} \tag{6.12}$$

where $f'_y(.)$, $f'_x(.)$, $g'_y(.)$, $g'_x(.)$ are the x and y derivatives of two mixture of Gaussians f and g, and $\alpha_\psi, \alpha_\theta \in [0, 360)$ is a random offset in direction; γ is used to manipulate the overall translation of minutiae.

We evaluate the performance of the above template transformation technique using the publicly available FVC 2002 database 2. We evaluate two different instances of the mixture of Gaussians based transformation with the values of γ being 30 (Trans-1) and 60 (Trans-2), respectively. Their respective transformation functions are shown in Figure 6.3. Figures 6.4-6.9 shows the evaluation measures described in Section 6.3 corresponding to these two instances of the mixture of Gaussians.

Figure 6.4 shows that the matching performance corresponding to the transformed template is significantly degraded compared to the original minutiae and the amount of degradation increases with γ. Also, the matching performance is lower when the attacker knows the key as shown by the ROC_{same} plots. As seen from Figures 6.5 and 6.6, the reduction in performance is mainly due to an increase in the FRR, which is primarily due to misalignment. The effect of misalignment is further exaggerated as a result of transformation. We use the high curvature points [91] in the fingerprint for pre-alignment, which may be erroneously extracted especially if the fingerprint



(a) (b) (c)

Figure 6.3: Minutiae transformation (a) minutiae distribution in the original image, (b) minutiae transformed according to mixture of Gaussians, where γ is 30, and (c) transformed minutiae when the value of γ is 60.

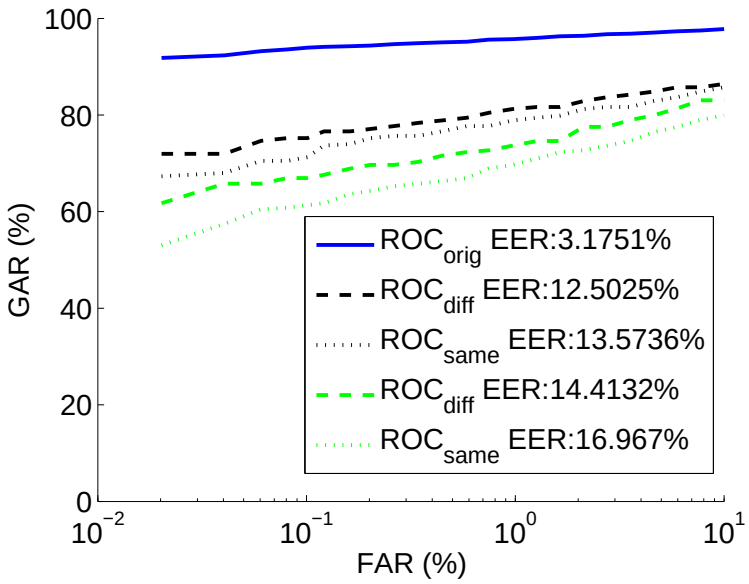


Figure 6.4: ROC_{orig}, ROC_{diff}, ROC_{same} for the mixture of Gaussian template transformation. Neurotechnology Verifinger 4.2 is used to perform minutiae matching. The evaluations in this figure and Figures 6.5, 6.6, 6.7, 6.8, 6.9, and 6.10 correspond to two transformations, Trans-1 and Trans-2, where γ equals 30 and 60, respectively. The curves corresponding to Trans-1 are shown in black where the curves corresponding to Trans-2 are shown in green.

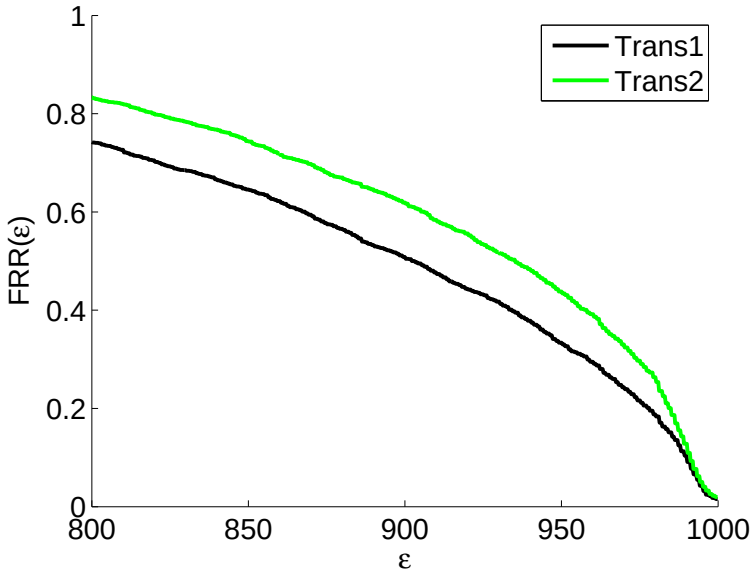


Figure 6.5: $FRR_T(\epsilon)$ for the mixture of Gaussian template transformation.

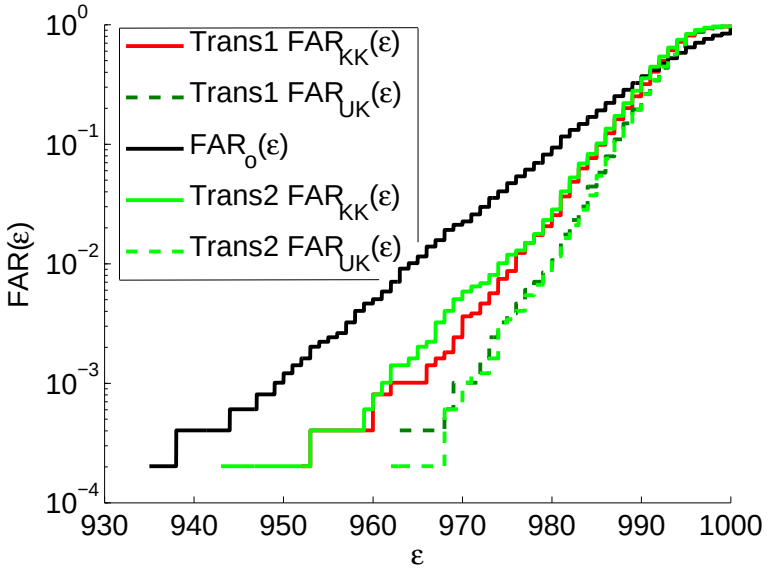


Figure 6.6: $FAR_{UK}(\epsilon)$ and $FAR_{KK}(\epsilon)$ for the mixture of Gaussian template transformation.

being processed is partial.

Figure 6.7 depicts the feasibility of intruding a different biometric system that has the same finger enrolled using the template inverted from the current system. At an operating threshold (ϵ) of 950, IRID for Trans-2 is around 51% when the attacker expends zero effort in inverting the template, i.e., $E(\beta) = 0$ or $\beta = E^{-1}(0)$. In other words, when the attacker just replays the most likely minutiae set from the pre-image of the transformed template without spending any effort on identifying the original minutiae from the pre-image, there is a 51% chance that he will succeed in intruding the system. A completely inverted template will further increase the intrusion rate to 54%. This value is even higher (64% for zero effort and 65% for full inversion) in the case of Trans-1. Note that the chances of intrusion increased only by 1% for Trans-1, while it increased by around 3% in case of Trans-2. This can be explained by the C-E curve shown in Figure 6.10; attacker can recover only 87% of minutiae without any effort in the case of Trans-2, whereas in the case of Trans-1 he can recover around 94%. Note that the value of $IRID(1, \epsilon)$ is upper bounded by $(1 - FRR_T(\epsilon))$, which corresponds to case where the attacker is able to exactly recover the original fingerprint.

Figure 6.8 shows the feasibility of successfully cross-matching two templates obtained from the same biometric trait but transformed using different transformation parameters. While both Trans-1 and Trans-2 have a zero cross match rate at $\epsilon = 950$, Trans-1 usually has slightly higher CMR_T than Trans-2. Figure 6.9 shows the ROC_{inv} corresponding for $\beta = E^{-1}(0)$. It shows that at a False Cross-match Rate of 0.1%, the chance of correctly linking the templates from two different systems is 91.5% for Trans-2 and 94% for Trans-1.

Table 6.2 tabulates the values of five security metrics (FRR_T, FAR_{UK}, FAR_{KK}, $IRID$, and CMR_T) for Trans-1 and Trans-2 at a threshold of $\epsilon = 950$. It is quite clear that while Trans-2 is more secure than Trans-1, it is less usable than Trans-1 because

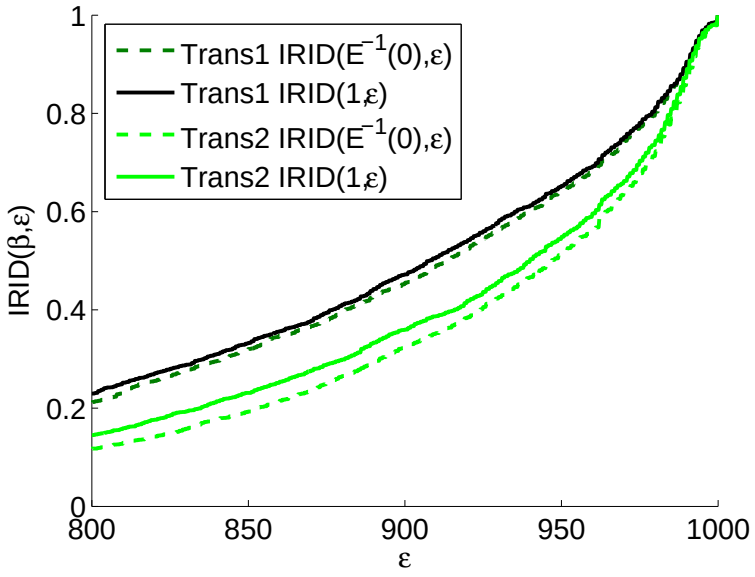


Figure 6.7: $IRID(\beta, \epsilon)$ for two different values of β for the mixture of Gaussian template transformation.

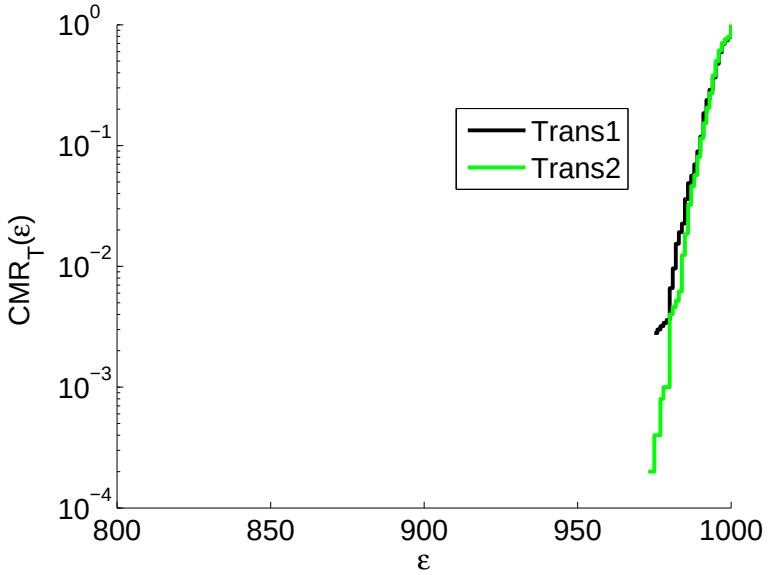


Figure 6.8: $CMR_T(\epsilon, \beta)$ at $\beta = 1$ for the mixture of Gaussian template transformation.

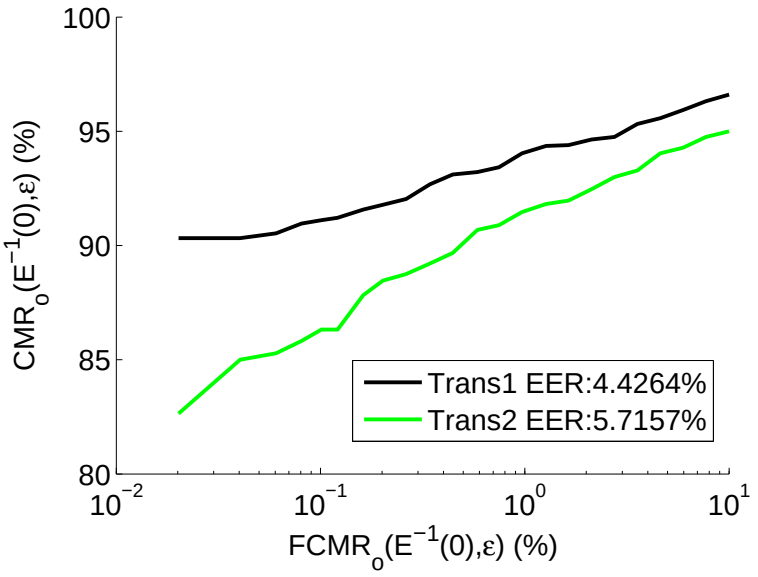


Figure 6.9: $\mathrm{ROC_{inv}}$ for the mixture of Gaussian template transformation.

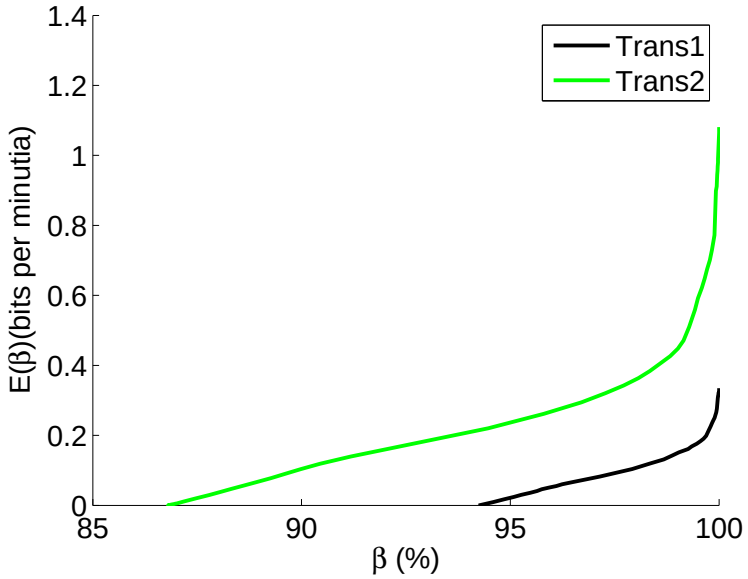


Figure 6.10: The C-E curve corresponding to two mixture of Gaussians based template transformations, Trans-1 and Trans-2, where γ equals 30 and 60, respectively.

of the higher false reject rate, which demonstrates the trade-off between security and usability that is a commonly encountered problem in biometric template protection. Moreover, our analysis shows that to prevent intrusion into other biometric systems that use the same trait and to mitigate linkage threats, it is not enough to design the transformation function such that it is computationally hard to recover the original template, but it must also be computationally difficult to obtain the pre-image of a transformed template. This issue has not received adequate attention in the literature [106].

6.5 Security of Biohashing Scheme

Biohashing is a vector based template protection technique that is used to secure different biometric traits such as fingerprints [123], face [122], palm [31], etc. In a typical biohashing scheme, the input biometric trait is represented as a vector of real numbers, say $\mathbf{x} \in \mathtt{R}^n$. This representation is then converted to a binary vector $\mathbf{b} = [b_1, b_2, ..., b_m]$ using the transformation matrix $M \in \mathtt{R}^{m \times n}$ and thresholds $\delta_i, i = 1, ..., m$. The biohash features are obtained as:

$$b_i = \begin{cases} 0 & \text{if } \sum_{j=1}^{n} M_{ij} x_j < \delta_i \\ 1 & \text{otherwise} \end{cases} \tag{6.13}$$

In our experiments, we use the FERET face database that contains 14,051 images. From these we select a subset of 500 subjects with two frontal images per subject. We align the images using the eye locations and crop a segment of size 100×125

Trans.	FRR_T	FAR_{UK}	FAR_{KK}	$IRID(E^{-1}(0), \epsilon)$	CMR_T
Trans-1	33%	0.02%	0.02%	64%	0%
Trans-2	44%	0.02%	0.02%	51%	0%

Table 6.2: Values of FRR, FAR_{UK}, FAR_{kk}, $IRID(E^{-1}(0), \epsilon)$, and CMR_T for the cancelable fingerprint template scheme corresponding to a threshold (ϵ) of 950.

from each image. Eigenface [128] features are used to represent the face images in our experiments. We use top 100 Eigenface features in order to extract 80 bits using the biohashing technique.

We now propose a method to recover a close approximation to the original biometric features given the biohash features ($\mathbf{b}$) and the transformation parameters, i.e., M and $\delta_i, i = 1, ..., m$. This problem can be formulated as an optimization problem as follows:

$$argmin \left|\left|\mathbf{x} - \mathbf{a}\right|\right|_2, \text{subject to} \tag{6.14}$$

$$\sum_{j=1}^{n} M_{ij}x_j < \delta_i, \text{ if } b_i = 0 \text{ and} \tag{6.15}$$

$$\sum_{j=1}^{n} M_{ij}x_j > \delta_i \text{ if } b_i = 1, \tag{6.16}$$

where $\mathbf{x}$ is the original biometric feature vector that is to be estimated, $\mathbf{b}$ is the vector of binary biohash features and $\mathbf{a}$ is one of the unrelated biometric feature vectors from a database. We use the lsqlin function available in the MATLAB optimization toolbox to obtain a solution to this problem. The above problem is solved for t different values of $\mathbf{a}$ in order to obtain $\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^t$. The final estimate of $\mathbf{x}$, $\hat{\mathbf{x}}$, is obtained as

$$\hat{\mathbf{x}} = \frac{\sum_{i=1}^{t} \mathbf{x}^i / d_i^2}{\sum_{i=1}^{t} 1/d_i^2}, \tag{6.17}$$

where d_i is the Hamming distance between biohash features corresponding to $\mathbf{x}^i$ and $\mathbf{a}^i$. The parameters $\mathbf{a}^i$'s are chosen such that Hamming distance between biohash features corresponding to a_i and b_i is less than certain threshold. Figure 6.11(b) shows an example of a face image reconstructed from the Eigenface features ($\hat{\mathbf{x}}$) that are estimated by inverting the biohash template ($\mathbf{b}$) using equations (6.14) and (6.17). We obverse that many distinctive features in the original face image (Figure 6.11(a))

are also present in the reconstructed image, which demonstrates the effectiveness of our inversion algorithm.

Figures 6.12-6.17 shows the evaluation of biohashing technique with respect to the different evaluation criteria proposed in Section 6.3 except the C-E curve, which is not directly applicable to biohashing. Figure 6.12 shows the three ROC curves i.e. ROC_{orig}, ROC_{same}, and ROC_{diff}. In contrast to the cancelable fingerprints technique, the ROC_{diff} of biohashing shows significantly better performance than ROC_{orig}, whereas ROC_{same} has lower matching performance compared to ROC_{orig}. This is because biohashing uses the external information (key or password) to significantly alter the distribution of the biometric features and increase the inter-user separation. However, this advantage is lost when the key is known to the adversary. On the other hand, the cancelable fingerprints scheme attempts to retain the fingerprint minutiae distribution, so that a traditional minutiae matcher can still be applied to match the transformed minutiae sets.

At the operating threshold of 20, the $IRID(E^{-1}(0), \epsilon)$ value is around 0.5, implying that the attacker has 50% success rate in intruding into a different system using the same biometric trait. The cross match rate in the transformed domain (CMR_T) is almost zero at the operating threshold of 20. As expected, the CMR_T follows FAR_{UK} closely. With respect to cross matching in the original domain, the CMR_O is around 82% at 10% $FCMR_O$ as shown in Figure 6.17. Table 6.3 lists the values of FRR, FAR_{UK}, FAR_{KK}, $IRID(E^{-1}(0), \epsilon)$, and CMR_T corresponding to the operating threshold of 20.

It is evident from Figure 6.15 that biometric templates from one database can be

FRR_T	FAR_{UK}	FAR_{kk}	$IRID(E^{-1}(0), \epsilon)$	CMR_T
9%	0.02%	5%	50%	0%

Table 6.3: Values of FRR, FAR_{UK}, FAR_{KK}, $IRID(E^{-1}(0), \epsilon)$, and CMR_T for the biohashing technique corresponding to a threshold (ϵ) of 20.

(a) (b)

Figure 6.11: Inversion of a biohash template. (a) Original face image from the FERET database (after alignment and cropping), (b) face image reconstructed from the Eigenface features ($\hat{\mathbf{x}}$) that are estimated by inverting the biohash template ($\mathbf{b}$) using equations (6.14) and (6.17).

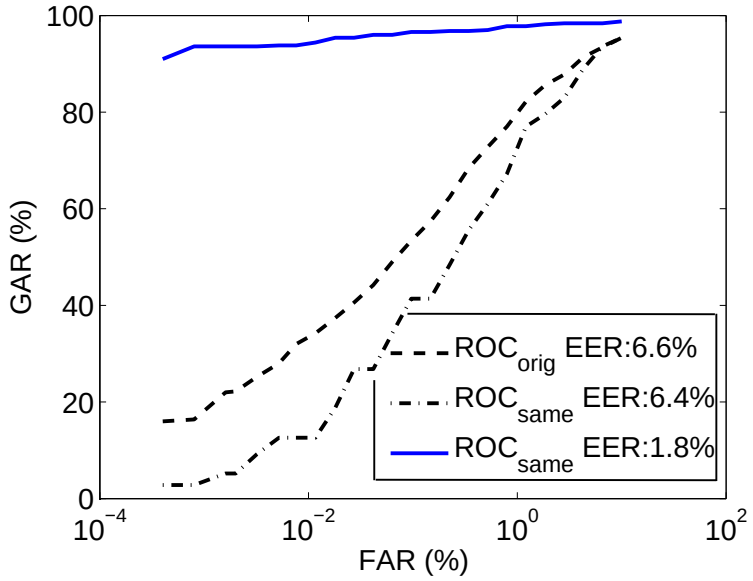


Figure 6.12: ROC_{orig}, ROC_{diff}, and ROC_{same} for biohashing technique. In this experiment, 100 Eigenface features were extracted and 80 bits/template were extracted using biohashing. The value of t used here is 100.

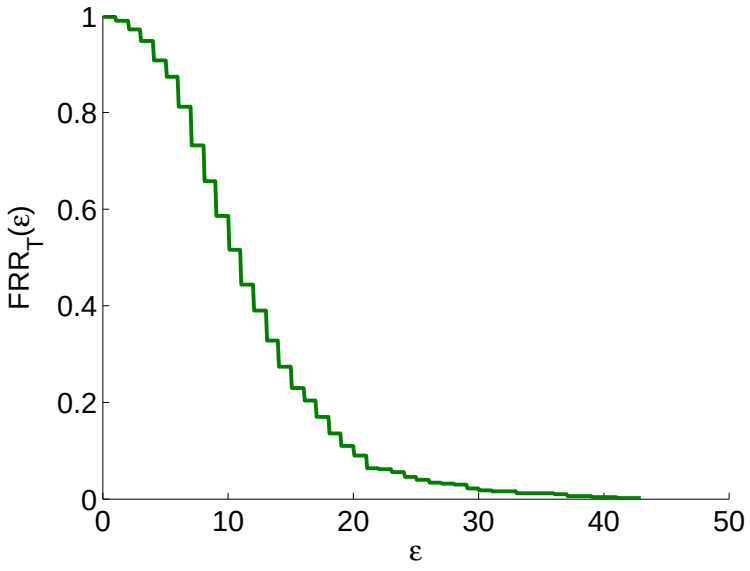


Figure 6.13: $FRR_T(\epsilon)$ for biohashing technique.

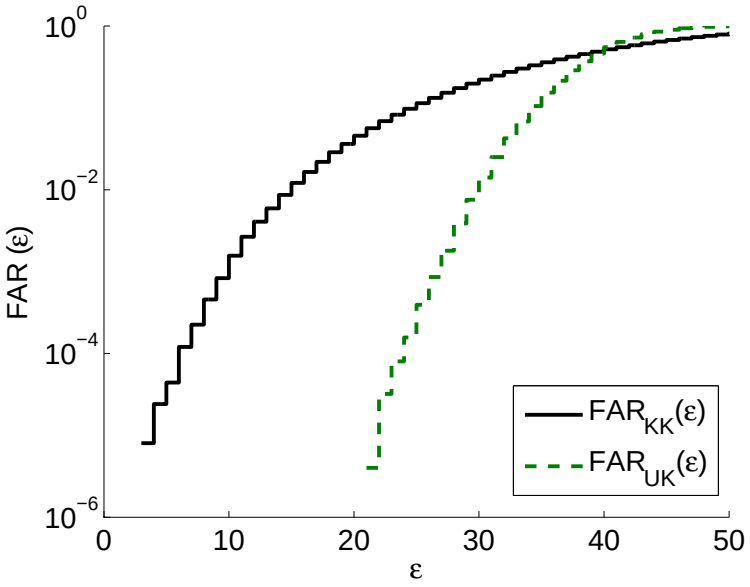


Figure 6.14: $FAR_{UK}(\epsilon)$ and $FAT_{KK}(\epsilon)$ for biohashing technique.

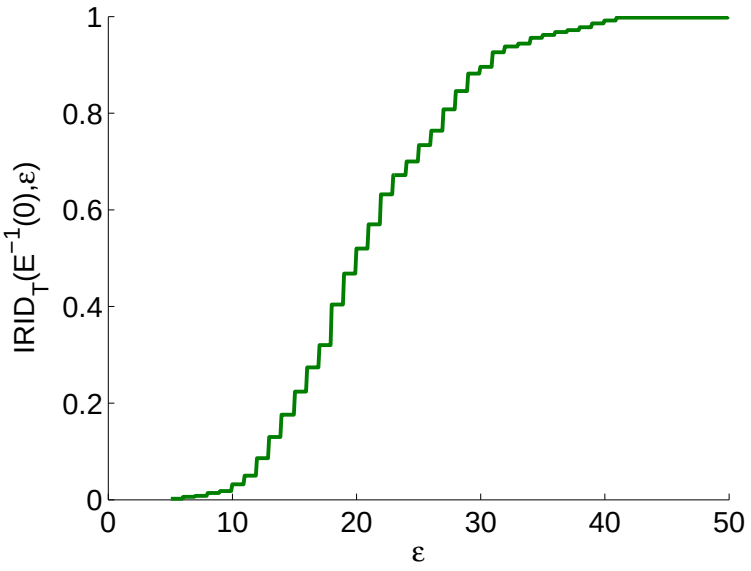


Figure 6.15: $IRID(\beta, \epsilon)$ for two different values of β for biohashing technique.

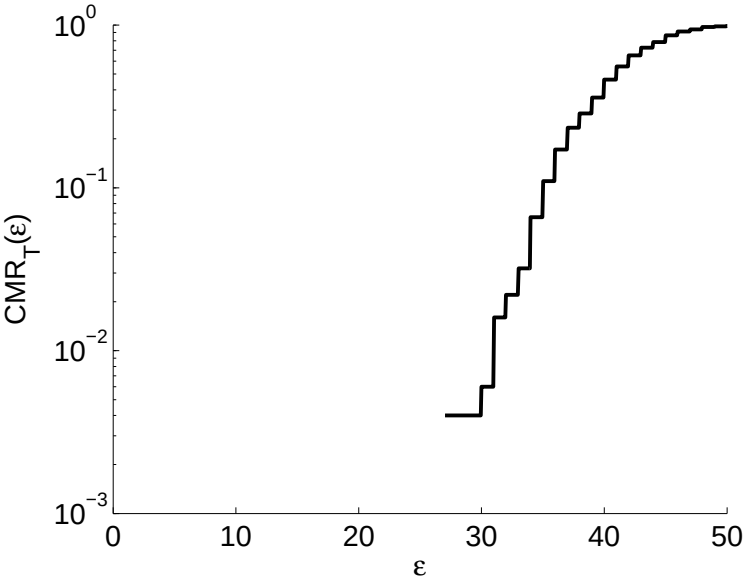


Figure 6.16: $CMR_T(\epsilon, \beta)$ for $\beta = 1$ for biohashing technique.

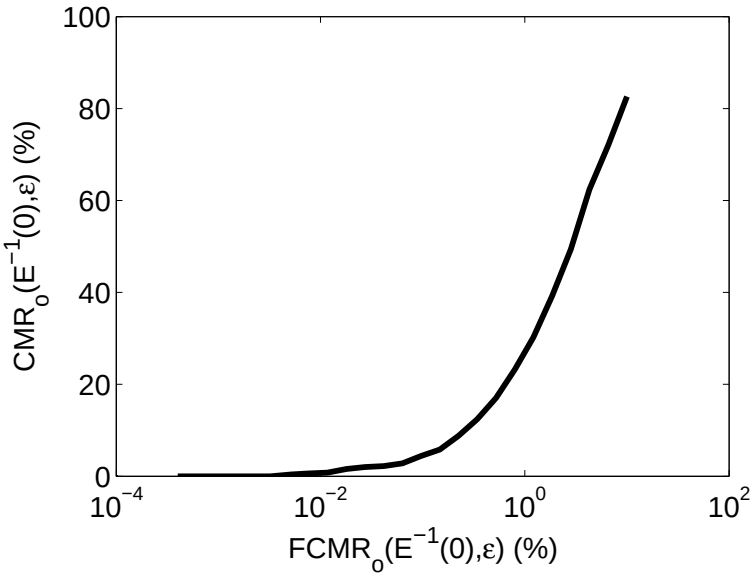


Figure 6.17: ROC_{inv} for biohashing technique.

inverted and used to compromise other systems using the same biometric trait. This is due to the contiguous nature of the pre-image of biohash. We propose a modification to the original biohashing scheme, which leads to a non-contiguous pre-image and thus is less vulnerable. The only difference between the modified and the original biohashing scheme is the binarization procedure. In the original biohashing technique, binarization is performed by first obtaining the median (δ) of each transformed feature and then thresholding the transformed features using this value. Instead, in the modified technique, each feature is thresholded at three different values: λ^{th}-, 50^{th}-, and $(100-\lambda)^{th}$- percentiles leading to four quanta for each feature. While the first and third quanta are represented as a 1, the other two quanta are represented as a 0. Note that $\lambda = 0$ leads to the original biohashing technique.

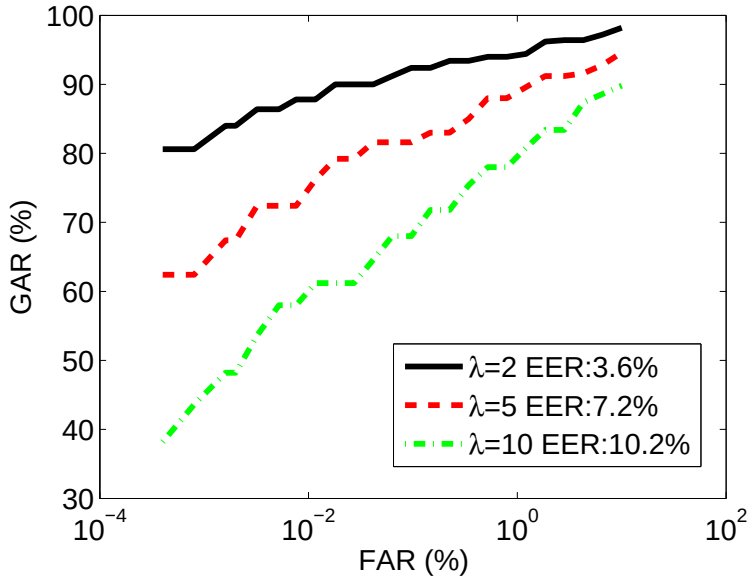
Figure 6.18 shows the ROC_{diff} corresponding to the modified technique for $\lambda \in \{2, 5, 10\}$. While there is certain reduction in the matching performance, it is now difficult to invert the template. The probability of guessing the correct quanta in each dimension is $p_\lambda = max(\lambda/50, 1-\lambda/50)$ given that one always chooses the larger

quanta. Thus if there are p Eigenface dimensions to be guessed using m biohash bits, the probability of identifying the correct quantum in which the non-quantized biohash values fall is p_λ^m. The security, in terms of bits, for guessing this is $-log_2(p_\lambda^m)$. In case $m = 80$, the security corresponding to $\lambda = 2, 5,$ and 10 are 4.7 bits, 12.1 bits, and 25.8 bits, respectively. However, in order to increase the security, m can be increased. In case $m = 400$, the security corresponding to $\lambda = 2, 5,$ and 10 is 23.6 bits, 60.8 bits, and 128.8 bits, respectively. ROC_{diff} corresponding to the modified biohashing scheme for different values of λ and m=80 and 400 are shown in Figure 6.18. The matching performance of the biohashing scheme reduces as λ is increased. However, increasing the number of dimensions improves the security as well as the matching performance in case the impostor does not know the key.

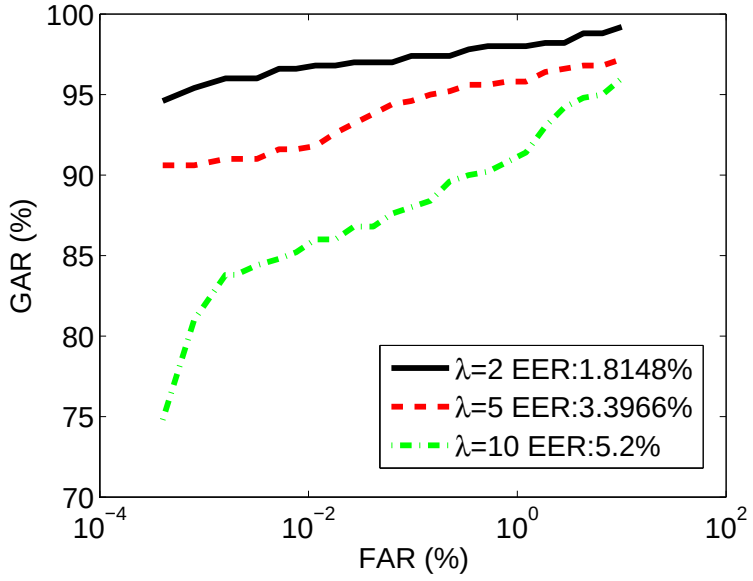
6.6 Summary

When a user's biometric template information falls into the hands of an adversary, it can seriously undermine the security (intrusion threats) of the biometric system and privacy (linkage threats) of the user. Hence, biometric template protection is a critical problem that needs to be addressed to enhance the public acceptance of biometric technology. Considering the recent surge in the number of techniques being developed for protecting the biometric templates, it is essential to develop a set of measures which can evaluate the strength of these techniques. One of the well known approaches for template protection is the template or feature transformation technique. Compared to biometric cryptosystems, template transformation schemes have certain advantages like easy revocability and flexibility in the matcher design. But these advantages are stymied by the lack of a thorough security analysis of these techniques.

We have proposed six different measures to evaluate the security strength of tem-



(a)



(b)

Figure 6.18: ROC_{diff} corresponding to the modified technique (a)ROC_{diff} for $\lambda \in \{2, 5, 10\}$ corresponding to the case when number of dimensions of PCA retained in 100 and number of bits extracted using biohashing technique is 80, and (b) shows the ROC_{diff} for $\lambda \in \{2, 5, 10\}$ corresponding to the case when number of dimensions of PCA retained is 500 and the number of bits extracted using biohashing technique is 400.

plate transformation schemes. Based on these measures, we analyze the security of two-well known transformation techniques, namely, cancelable fingerprints and biohashing. Our analysis shows that both these techniques are vulnerable to intrusion and linkage attacks, as indicated by their high $IRIS$, $IRID$ and CMR_O values. In particular, the vulnerability of the biohashing scheme is due to the relative ease with which an impostor can invert the transformed template to obtain a close approximation to the original biometric template. Hence, we propose a modification to the biohashing scheme that can address this limitation, though at the expense of a marginal reduction in the matching performance.

In the case of cancelable fingerprint template scheme, the vulnerabilities arise because an impostor can easily obtain the pre-image of the transformed template. Even though it is computationally hard to recover the original template from the pre-image, the pre-image itself is sufficient to carry out linkage and intrusion attacks. Therefore, for enhanced template security, we argue that the non-invertibility of a transformation function must also be measured in terms of the complexity of obtaining the complete pre-image of a transformed template, rather than simply analyzing the complexity of recovering the original template. However, proving the computational hardness of this problem is not easy because it may be possible to design greedy algorithms that can perform the inversion efficiently.

Our experiments also highlight the well-known tradeoff between the security and usability. In this context, hybrid biometric cryptosystems may have an edge because the complementary strengths of template transformation and biometric cryptosystems can be leveraged to improve both the security and usability of a biometric system.

Chapter 7

Summary and Future Research

7.1 Summary

With the proliferation of biometric recognition systems in commercial sector, security of the stored biometric data is increasingly becoming crucial. As assessed in this dissertation, current biometric systems have a number of vulnerabilities and a motivated adversary can undoubtedly cause severe harm to a biometric system as well as the users enrolled in the system. Furthermore, due to the permanent nature of biometrics data its theft and misuse may be irreparable. If someone's fingerprints or iris patterns are stolen and are falsely linked to high susceptibility of a dreaded disease, the person may be unable to obtain a medical insurance. Stolen biometric data may devoid a person of any conveniences offered by the biometric systems due to the concern of being easily impersonated using spoof biometrics. While these threats may not appear to be imminent, the pace at which biometric systems are proliferating, the wealth of information one may harness by staging extensive theft of biometric data would definite motivate the con men. Through this dissertation, we have provided a comprehensive set of tools that we hope would be instrumental in circumventing any compromises of the biometric systems and in maintaining public trust in using

biometric systems.

The first chapter of this dissertation details various aspects of biometric system security. A designer of a biometric system may use this discussion as reference while building a biometric system that is robust to any theft or sabotage. The second chapter discusses the vulnerability of current biometric data storage format and shows that even some of the recently proposed formats for biometric templates can be easily used to recover the biometric image and thus construct the spoof biometrics. The third chapter analyzes the construction and security analysis of fuzzy vault and fuzzy commitment, two of the most common biometric cryptosystems. The comprehensive security analysis allows use of a single GAR-security curve in order to assess both the security as well as usability aspects of a biometric cryptosystem. The fourth chapter develops techniques to combine multiple non-homogeneous biometric templates in a biometric cryptosystem. The developed technique shows a significant improvement in terms of security as well as matching accuracy compared to the individual biometric traits. One of the limitations of a multibiometric cryptosystem is that only a subset of biometric traits are required to decode the protected template which would reveal all the biometric templates used in the cryptosystem. A constrained template security system was developed to overcome this limitation. The fifth chapter proposes two new improvements to a fingerprint fuzzy vault: incorporation of user password and inclusion of minutia descriptors while constructing the vault. Incorporating user passwords into a fuzzy vault allows two factor authentication thereby significantly improving the security as an impostor would be required to provide both biometric data as well as the correct password. Incorporating minutiae descriptors into the fuzzy vault significantly improves the matching accuracy. Note that one of the major hurdles in the acceptance and implementation of biometric template protection techniques is their lower matching accuracy compared to the normal biometric systems. The sixth chapter provides a detailed analysis of the template transformation

techniques and studies two common examples of the transformation techniques. This chapter provides a list of evaluation metrics that can be used to formally evaluate and compare the security and usability of a template transformation technique. Note that proper evaluation is essential to motivate development and acceptance of good security techniques [10].

7.2 Future work

We suggest the following tasks as future work that would significantly improve the security of biometric systems.

- A number of secure biometric recognition protocols based on homomorphic encryption techniques have been proposed. A thorough security analysis of these protocols is needed.

- We presented a technique to recover the fingerprint image given MCC descriptor only template. An extensive analysis of representations of other biometric traits as well as other representations of fingerprint needs to be conducted.

- In chapter five, we provide a technique to incorporate password into a fingerprint fuzzy vault. This technique can be further generalized to fuzzy commitment as well. A formal analysis of the optimality of this technique is also desired.

- We proposed a technique to incorporate minutiae descriptors into fuzzy vault. Techniques may also be developed to incorporate information regarding global fingerprint pattern such as ridge orientation field and ridge frequency map.

- One of the crucial elements in the analysis of template transformation techniques is the design of a template inversion technique. We have proposed inversion techniques for cancelable fingerprints and biohashing. Inversion techniques for

other available template transformation techniques may be designed as future work. This may also entail formal analysis and categorization of developed inversion techniques.

Appendices

Appendix A

Entropy of Biometric features

By entropy of biometric features, we mean the minimum average number of bits required to represent a biometric feature vector. A simple approximation of the entropy for iriscodes was provided by Daugman [35], as

$$N_* = p(1-p)/\sigma^2 \tag{A.1}$$

where p is the mean value of the observed normalized Hamming distances corresponding to impostor matches and σ^2 is their variance. This estimation assumes that biometric features consists of a set of Bernoulli random variables, which are independent and identically distributed (with uniform distribution). In our case, since the mean normalized Hamming distance was less than 0.5, we assume that few bits are constant for all biometric samples. Normalized Hamming distance (ρ_{NH}) is thus computed as

$$\rho_{NH} = \rho_H/(2 * \mu) \tag{A.2}$$

where μ is the mean of the impostor Hamming distances and ρ_H is the corresponding original Hamming distance. This value of normalized Hamming distance (ρ_{NH}) is

used to computed the values of p and σ which, in turn, is used to estimate the entropy of biometric features using eq. (A.1).

Appendix B

Inversion of Cancelable Fingerprint

As noted in [10], proper evaluation is essential to motivate the development and acceptance of good security techniques. Here, we present a measure of non-invertibility for cancelable fingerprint templates while assuming that the user specific key is known to the adversary. The proposed technique measures the relationship between the number of guesses (*effort*) required by an adversary to recover a certain fraction (*coverage*) of the biometric template given the transformed template. The different (coverage-effort)-tuples are plotted to obtain the Coverage-Effort (CE) curve. The computation of a CE curve consists of three main steps:

1. Pre-image Computation: Compute the pre-images of each transformed minutia such that transformation of all the pre-image minutiae would lead to the given transformed minutia.

2. Minutiae Likelihood Computation: Estimate the relative probability of each of the minutiae in the pre-image using kernel density estimation.

3. Non-invertibility Measure Computation: Sort the pre-images according to their likelihoods and compute the coverage i.e. the number of true pre-images guesses given that the adversary checks only a certain portion of the pre-images.

We note that the proposed measure is sufficiently generic to be useful for any minutiae transformation technique such that the transformation can be evaluated at any given point and is piecewise differentiable.

B.1 Minutiae Template Transforms

A minutiae based fingerprint template, say T, consists of a collection of n minutiae i.e. $T = \{(x_1, y_1, \theta_1), (x_2, y_2, \theta_2), ..., (x_n, y_n, \theta_n)\}$. The transformation function considered here, $\phi(.)$, takes T to another set of n minutiae i.e. $\phi(T) = \{(x'_1, y'_1, \theta'_1), (x'_2, y'_2, \theta'_2), ..., (x'_n, y'_n, \theta'_n)\}$.

A desirable transformation should account for the intra-class variation while at the same time providing a reasonable template security. A number of minutiae based template transformation techniques have been proposed (see [73, 92, 106]) where the configuration of each minutia is changed according to a user specific key to obtain the transformed template. Ratha et al. [106] proposed three different kinds of transformations i.e. cartesian, polar, and functional as illustrated in Figure 6.2. The many-to-one nature of these transforms provides non-invertibility even for the case when the adversary knows the user specific key. A cartesian transformation tessellates the image plane into rectangles and then shuffles the rectangles based on the user password such that any two rectangles can map on to a single rectangle. Instead of rectangles, a polar transform tessellates the image plane into sections of annular regions around a center point. A functional transformation or the mixture of Gaussians based transform, however, transforms the minutiae based on a function evaluated over a minutiae configuration.

B.1.1 Mixture of Gaussians based Transform

Due to its generic nature and acceptable performance [106], we use the functional transformation technique based on a mixture of Gaussians to compute a measure of non-invertibility. In order to transform a minutia, functions consisting of a mixture of Gaussians and its derivatives are evaluated at the position of minutia and then the minutia is translated according to the values obtained. For the sake of simplicity, we restrict the transformation function to change only the x and y coordinates of a minutia.

The mixture of Gaussians used to obtain the transformation function is given by:

$$f(\vec{x}) = \sum_{i=1}^{K} t_i \pi_i e^{-\frac{1}{2}(\vec{x}-\vec{\mu_i})\Sigma_i^{-1}(\vec{x}-\vec{\mu_i})'} \tag{B.1}$$

where K is the number of components, and π_i, t_i, μ_i, and Σ_i correspond to the mixing probabilities, the signs (+ or -), means, and covariance matrices of the different components, respectively. $\vec{x}$ is a vector representation of a minutia consisting of only the x and y coordinates of the minutiae. In our experiments, where the fingerprints are captured at 569 ppi resolution and are 560×296 in size, K is taken to be 24, Σ_i is taken to be a diagonal matrix with each diagonal entry equal to 50^2 for each component. The remaining parameters are determined using the user specific key.

The transformation of each minutia is represented as direction of minutia translation (denoted by ϕ_θ) and magnitude of minutia translation (denoted by ϕ_d). The two components of the transformation can be obtained as:

$$\phi_\theta(\vec{x}) = \arctan\left(\frac{f'_y(\vec{x})}{f'_x(\vec{x})}\right) + \alpha, \tag{B.2}$$

$$\phi_d(\vec{x}) = \gamma \left\{ 1 + \left[\sum_{i=1}^{K} t_i \pi_i e^{-\frac{1}{2\sigma^2}(\vec{x}-\vec{\mu})(\vec{x}-\vec{\mu})'} \right] \right\}, \tag{B.3}$$

where $f'_y(.)$ and $f'_x(.)$ are the x and y derivatives of f and $\alpha \in [0, 360)$ is a random offset in direction; γ is used to manipulate the overall translation of minutiae. Figure 6.3 shows the fingerprint minutiae transformed according to the functional transformation generated using different values for γ (30 and 60).

B.2 Non-invertibility Measure

We propose a three-stage procedure for estimating the non-invertibility that involves: i) pre-image identification, ii) pre-image likelihood evaluation, and iii) non-invertibility measure computation.

B.2.1 Pre-image Computation

In order to compute the pre-image of a minutia, all 4-pixel neighborhoods of the form $(i, j), (i + 1, j), (i, j + 1), (i + 1, j + 1)$ from the original fingerprint image space are transformed and the ones that *cover* a particular transformed minutia are used to obtain candidate pre-images of that minutia. Any one out of the four points in the covering neighborhood is taken as the pre-image minutia. If multiple pre-image points are sufficiently close to each other, only one of them is included in the pre-image set. Complete link clustering [60] is used for this purpose with a splitting criteria depending on the precision required in the guessed pre-image. An extension to incorporate change in θ will involve an 8-point 3D neighborhood including θ instead of a 2D neighborhood. In some cases depending on the transform, if the 4-pixel neighborhood is severely distorted, certain pre-images might not be detected. Such cases will, however, not arise if the pre-image is computed as a closed form solution or a sufficiently fine grid is used.

B.2.2 Pre-image Likelihood Computation

Let $\vec{v}$ be a transformed minutia and $\vec{u}^1, \vec{u}^2, ..., \vec{u}^m$ be the m pre-images of $\vec{v}$ under the transformation ϕ. Further, let $l_v \in 1, 2, ..., m$ be a random variable indicating which of the pre-images of $\vec{v}$ is the true one. We are interested in computing the probability $P(l_v = r|\vec{v} = \vec{a} = (x_v, y_v, \theta_v))$. Using the Bayes theorem,

$$P(l_v = r|\vec{v} = \vec{a}) = \frac{p(\vec{v}{=}\vec{a}|l_v{=}r) * P(l_v{=}r)}{\sum_{i=1...m} p(\vec{v}{=}\vec{a}|l_v{=}i) * P(l_v{=}i)}. \tag{B.4}$$

Taking the prior probability $P(l_v = i) = 1/m, \forall i = 1, 2, ..., m$ (no preference for any particular pre-image) and converting $p(\vec{v} = \vec{a}|l_v = r)$ to $p(\vec{u}^r)$,

$$P(l = r|\vec{v} = \vec{a}) = \frac{p(\vec{u}^r)/J_\phi(\vec{u}^r)}{\sum_{k=0,...,m-1} p(\vec{u}^k)/J_\phi(\vec{u}^k)}, \tag{B.5}$$

where $J_\phi(\vec{u}^k)$ is the Jacobian (cf. [96], page 234) of the transformation ϕ which can be computed either numerically or in a functional form depending on the complexity of ϕ.

In order to compute $p(\vec{u}^r)$, we perform a kernel density estimation of minutiae represented as the (x, y, θ)-tuple using a Gaussian kernel with a leave-one-out estimate of the bandwidth[1]. Before estimating the probability density, we align all the fingerprints using their high curvature points based on the Trimmed Iterative Closest Point (ICP) algorithm [91]. Note that an alignment of fingerprints prior to density estimation leads to a more distinctive probability density with a low entropy. Figure B.1 shows the estimated probability density.

[1] We use the Kernel Density Estimation Toolbox for Matlab provided by Alexander Ihler (Available at: http://www.ics.uci.edu/ ihler/code/kde.html).

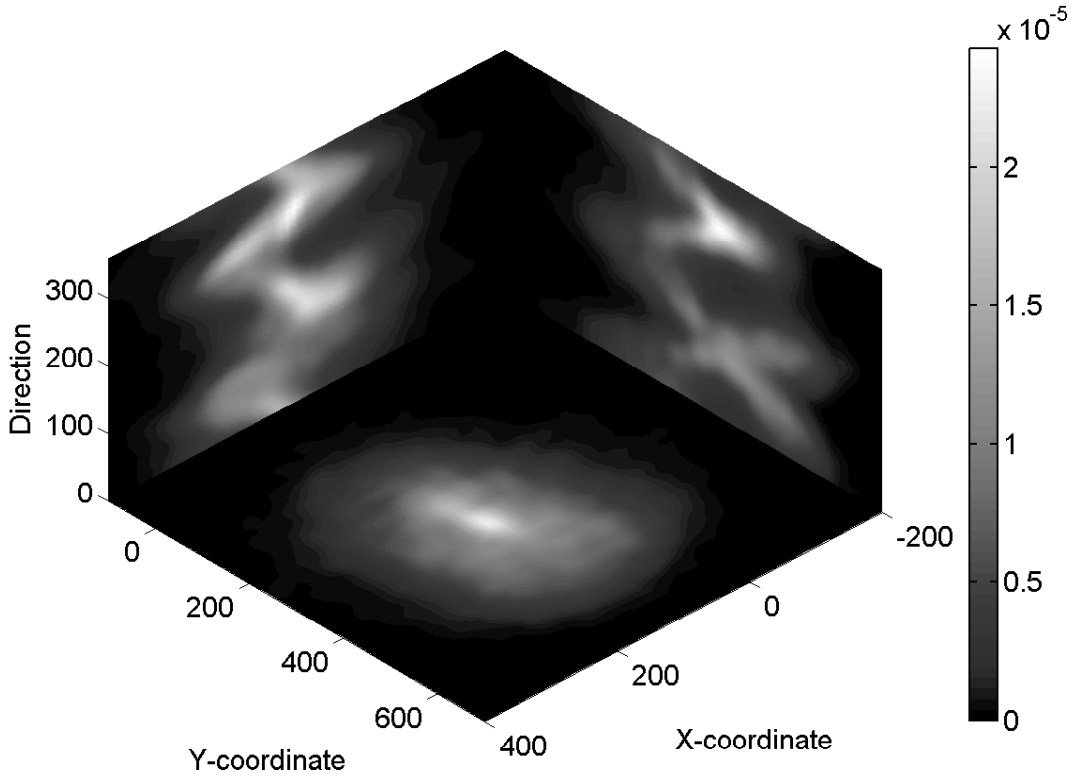


Figure B.1: Marginal densities of minutiae in (x, y), (x, θ), and (y, θ) planes.

B.2.3 Non-invertibility Measure Computation

We compute a measure of non-invertibility as the number of computations required by an adversary to guess the original minutiae set using a specific attack strategy. Let there be n different minutiae in the transformed template whose pre-image needs to be computed. An attack strategy includes the order in which an adversary guesses the various n-tuples corresponding to the selection of a particular pre-image for each of the n minutiae. Note that if there are m_i pre-images of the i^{th} minutia then the number of n-tuples that the adversary needs to prioritize is $\Pi_{i=1,..,n} m_i$ which could be very large. In order to make the analysis feasible, we assume that instead of guessing from all the pre-images of a minutia, the adversary guesses only from some of the more probable pre-images of each minutiae. In the limiting case, the adversary will just select the most-probable pre-image for each minutiae.

In our experiments, we consider an adversary that checks only the 2^{H_i} most probable pre-images[2] of the minutia $v_i, i = 1...n$. Here H_i is the entropy or the difficulty in guessing the true pre-image given by

$$H_i = -\sum_{r=1}^{m_i} P(l_{v_i} = r|\vec{v_i}) \log_2(P(l_{v_i} = r|\vec{v_i})), \tag{B.6}$$

where m_i is the number of pre-images of v_i. In this scenario, $\Pi_i 2^{H_i}$ different guesses will be made simultaneously for each individual minutia leading to an effort equivalent to $1/n \sum_i H_i$ bits per minutia. The corresponding coverage is computed as the fraction of minutiae whose true pre-images lie among the searched space. Note that these two values, i.e. effort and coverage, provide only a single point on the Coverage-Effort curve. In order to increase or decrease the coverage, we assume that adversary searches for $min(m_i, \lceil 2^{H_i+\eta} \rceil)$ most probable pre-images per minu-

[2]Note that for a random variable Z with m equally likely pre-images, $m = 2^{H_Z}$ where H_Z is its entropy.

tia, where $\eta \in [-max(H_i), max(H_i)]$. Note that in this case, the adversary is making $\approx 2^{n\eta}$ times more (or less if η is negative) guesses than the previous case. This leads to the complete CE curves as shown in Figure B.2.

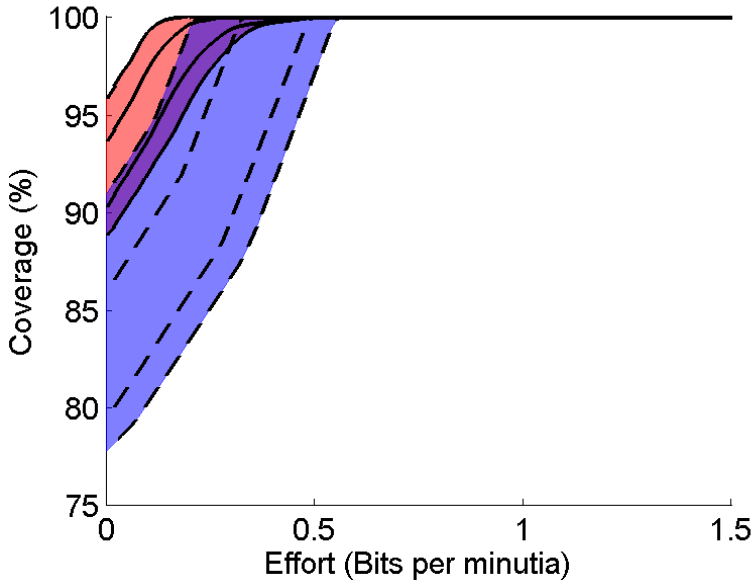
B.3 Experiments

To demonstrate the effectiveness of the proposed non-invertibility measure, we evaluated it on the publicly available FVC2002 database-2 which contains 800 fingerprint images (100 fingers $\times$ 8 impressions/finger) of size 560×296 captured at 569 ppi resolution. There are about 35 minutiae per fingerprint in the database. The experiments are based on mixture of Gaussians based functional transformation technique.

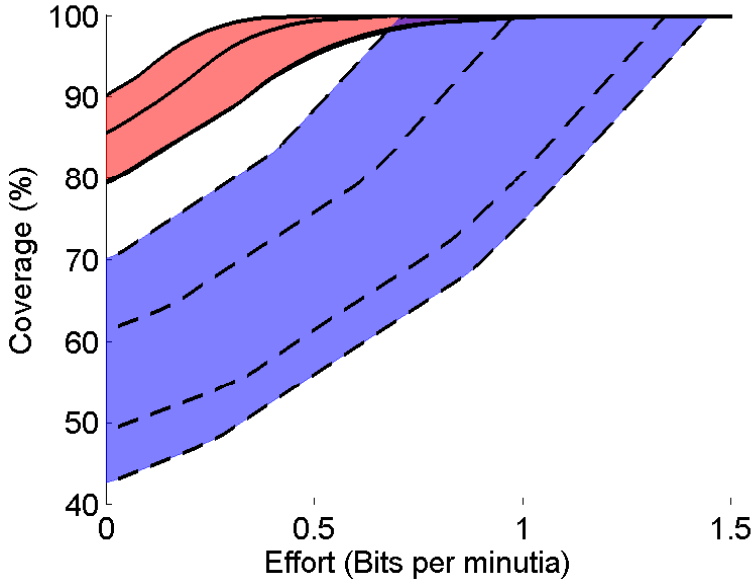
Figure B.2 shows the Coverage-Effort curves corresponding to the mixture of Gaussians based transformation with two different parameter settings. For each parameter setting, four different randomly generated transformation instances were used, say corresponding to using four different passwords. We also obtain the CE curves corresponding to the case when the minutiae distribution is uniform. As shown in Figure B.2, the curves obtained using the uniform minutiae distribution depict significantly greater security as compared to when the true minutiae distribution is taken into consideration. This is due to the fact that the minutiae with low pre-image entropy have the correct pre-image among the first few highly probable pre-images. Also, it can be observed that different parameter values can lead to significantly different security for a transformed template. Note that the proposed approach can be used to compute the coverage effort curve for individual fingerprints. Figure B.3 shows the CE curve and the corresponding minutiae from a fingerprint.

We used the Neurotechnology Verifinger SDK [95] in order to perform the minutiae matching. The genuine matches were performed by matching each of the eight impressions of a finger with each other impression leading to 2,800 genuine matches and

the impostor matches were performed by matching the first impression of each finger with the first impression of the remaining fingers leading to 4,950 impostor matching scores. The matching results reported here are for the case when the impostor knows the true user specific key i.e. all the templates in the database have been transformed using the same user specific key. Figure B.4 shows the ROC curves corresponding to the transformed templates based on two different parameter settings of the mixture of Gaussians transform (same as those used in computing the CE curves). It can be observed that the parameter setting that leads to lower security has better matching performance verifying the trade-off between security and matching performance as expected.

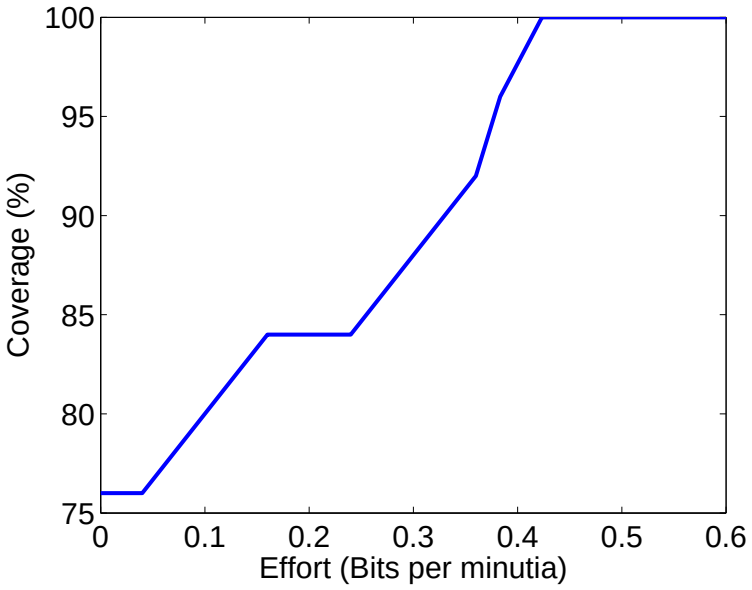


(a)

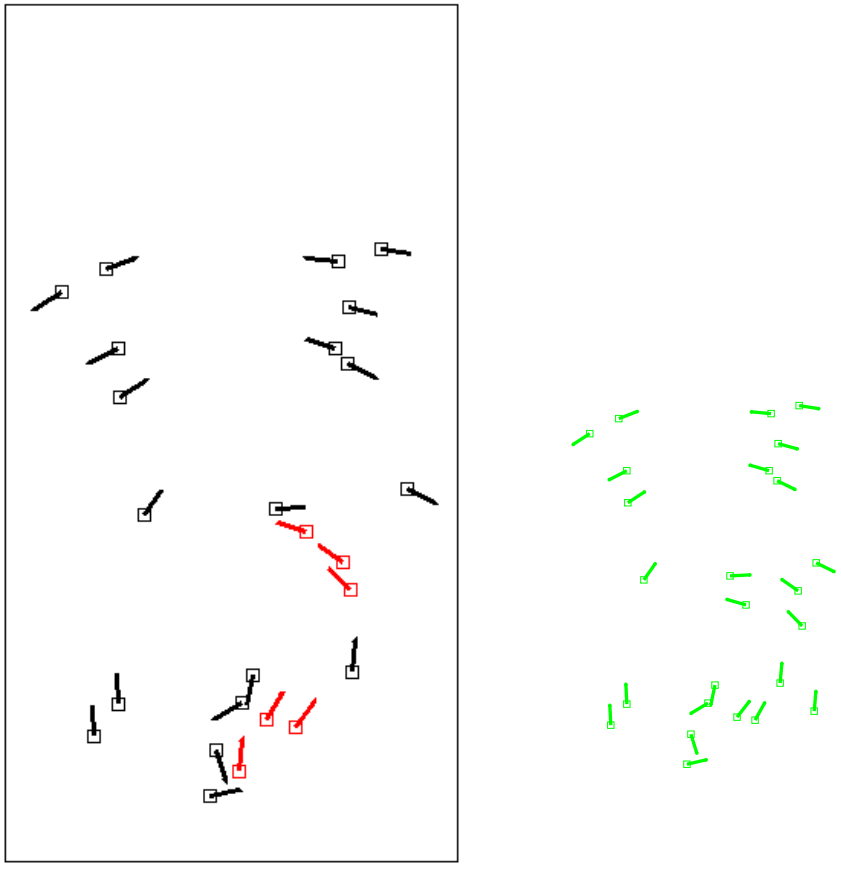


(b)

Figure B.2: Coverage-Effort curves for the mixture of Gaussians based feature transformation. (a) and (b) CE curves for the case when γ equals 30 and 60, respectively keeping the remaining parameters fixed. In each figure four different instances of the transformation are shown with four different solid lines. The dotted lines correspond to random guesses of the true pre-image. The size of the colored regions indicate variance in the security imparted by different instances of the transform.



(a)



(b) (c)

Figure B.3: CE curve for individual finger. (a) shows the CE curve, (b) the most likely pre-image of each minutia with the correctly guessed minutiae shown in black, and (c) the true pre-images with the total number of pre-images per minutia.

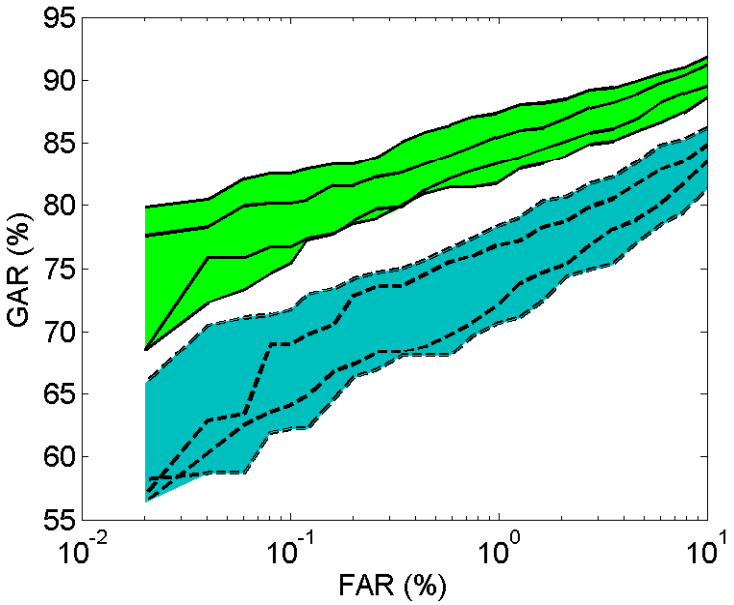


Figure B.4: ROC curves for the mixture of Gaussians based transformation of fingerprint template. Four random instances of the two cases where γ (see Eq. (B.3)) equals 30 and 60 are shown as solid and dotted lines, respectively. The size of colored regions indicate variance in performance of different instances of the transform.

Bibliography

Bibliography

[1] Bertillonage. http://www.nlm.nih.gov/visibleproofs/education/measure/.

[2] Privaris inc. http://www.privaris.com/.

[3] FaceIT SDK, L-1 Identity Solutions.

[4] FaceVacs SDK, Cognitec Systems. http://www.cognitec-systems.de/FaceVACS-SDK.19.0.html.

[5] XM2VTS face database. http://www.ee.surrey.ac.uk/CVSSP/xm2vtsdb/.

[6] Advanced Encryption Standard. National Bureau of Standards, U.S. Department of Commerce, Washington D.C., November 2001.

[7] A. Adler. Sample Images can be Independently Restored from Face Recognition Templates. In *Proceedings of Canadian Conference on Electrical and Computer Engineering*, volume 2, pages 1163–1166, Montreal, Canada, May 2003.

[8] A. Adler. Vulnerabilities in Biometric Encryption Systems. In *Proceedings of Audio- and Video-Based Biometric Person Authentication*, pages 1100–1109, 2005.

[9] N. G. Altman. Palmprint identification system. US Patent No. 3,581,282, 1971.

[10] A. Anderson and T. Moore. Information Security: where computer science, economics and psychology meet. *Philosophical Transactions of the Royal Society A*, 367(1898):2717–2727, July 2009.

[11] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *IEEE Symposium on Foundations of Computer Science*, pages 459–468, 2006.

[12] A. Antonelli, R. Cappelli, D. Maio, and D. Maltoni. Fake Finger Detection by Skin Distortion Analysis. *IEEE Transactions on Information Forensics and Security*, 1(3):360–373, September 2006.

[13] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces versus Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(7):711–720, 1997.

[14] E. R. Berlekamp. *Algebraic Coding Theory*. McGraw Hill, 1968.

[15] P. Besl and N. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[16] W. W. Bledsoe. Man-machine facial recognition: Report on a large-scale experiment. Technical Report 22, Panoramic Research, Inc., Palo Alto, California, 1966.

[17] A. C. O. Bock. Automatic fingerprint machine. Fingerprint Machine Corporation, New York, US Patent No. 1,529,864, 1925.

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

[19] J. Bringer and H. Chabanne. An authentication protocol with encrypted biometric data. In *Proceedings of the Progress in Cryptology, AFRICACRYPT*, pages 109–124, 2008.

[20] J. Bringer, H. Chabanne, G. Cohen, B. Kindarji, and G. Zemor. Theoretical and practical boundaries of binary secure sketches. *IEEE Transactions on Information Forensics and Security*, 3:673–683, 2008.

[21] J. Bringer and V. Despiegel. Binary feature vector fingerprint representation from minutiae vicinities. In *Proc. IEEE 4th International Conference on Biometrics: Theory, Applications, and Systems*, Crystal City, September 2010.

[22] W. E. Burr, D. F. Dodson, and W. T. Polk. Information Security: Electronic Authentication Guideline. Special Report 800-63, NIST, April 2006.

[23] R. Cappelli, M. Ferrara, and D. Maltoni. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(12):2128 – 2141, 2010.

[24] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni. Fingerprint Image Reconstruction From Standard Templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1489–1503, 2007.

[25] E-C. Chang, R. Shen, and F. W. Teo. Finding the original point set hidden among chaff. In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, June 2009.

[26] C. Chen and R. Veldhuis. Binary Biometric Representation through Pairwise Polar Quantization. In *Proc. International Conference on Biometrics*, pages 72–81, 2009.

[27] C. Chen, R. N. J. Veldhuis, T. A. M. Kevenaar, and A. H. M. Akkermans. Biometric Quantization through Detection Rate Optimized Bit Allocation. *EURASIP Journal on Advances in Signal Processing*, 2009.

[28] X. Chen, J. Tian, and X. Yang. A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure. *IEEE Trans. Image Processing*, 15(3):767–776, 2006.

[29] S. Cimato, M. Gamassi, V. Piuri, R. Sassi, and F. Scotti. Privacy-aware biometrics: Design and implementation of a multimodal verification system. In *Proc. IEEE Annual Conference on Computer Security Applications*, Los Alamitos, CA, 2008.

[30] Cogent. Cogent fusion. `http://www.cogentsystems.com`.

[31] T. Connie, A. B. J Teoh, M. Goh, and D. C. L Ngo. PalmHashing: a novel approach for cancelable biometrics. *Information Processing Letters*, 93(1):1–5, 2005.

[32] S. Crihalmeanu, A. Ross, S. Schuckers, and L. Hornak. A protocol for multibiometric data acquisition, storage and dissemination. Technical report, Lane Department of Computer Science and Electrical Engineering, WVU, 2007.

[33] J. Daugman. Biometric personal identification system based on iris analysis. U.S. Patent No. 5,291,560, 1994.

[34] J. Daugman. Recognizing Persons by their Iris Patterns. In A. K. Jain, R. Bolle, and S. Pankanti, editors, *Biometrics: Personal Identification in Networked Society*, pages 103–122. Kluwer Academic Publishers, London, UK, 1999.

[35] J. Daugman. The importance of being random: statistical principles of iris recognition. *Pattern Recognition*, 36:279–291, 2003.

[36] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[37] C. Fang, Q. Li, and E.-C. Chang. Secure Sketch for Multiple Secrets. In *Proc. of ACNS*, 2010.

[38] F. Farooq, R.M. Bolle, T.Y. Jea, and N. Ratha. Anonymous and revocable fingerprint recognition. In *Proc. IEEE Computer Vision and Pattern Recognition*, June 2007.

[39] H Faulds. On the skin-furrows of the hand. *Nature*, 22:605, 1880.

[40] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Proceeding of IEEE Computer Vision and Pattern Recognition*, pages 524–531, 2005.

[41] J. Feng. Combining minutiae descriptors for fingerprint matching. *Pattern Recognition*, 41(1):342–352, 2008.

[42] J. Feng and A. K. Jain. Fingerprint reconstruction: From minutiae to phase. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):209–223, February 2011.

[43] J. Feng and J. Zhou. A performance evaluation of fingerprint minutia descriptors. In *Proc. International Conference on Hand-based Biometrics*, Hong Kong, 2011.

[44] Y. C. Feng and P. C. Yuen. Protecting Face Biometric Data on Smartcard with Reed-Solomon Code. In *Proceedings of CVPR Workshop on Biometrics*, page 29, New York, USA, June 2006.

[45] Y. C. Feng, P. C. Yuen, and A. K. Jain. A hybrid approach for generating secure and discriminating face template. *IEEE Transactions on Information Forensics and Security*, 5(1):103–117, 2010.

[46] Y. C. Feng, P.C. Yuen, and A.K. Jain. A hybrid approach for face template protection. In *Proceedings of SPIE Conference on Biometric Technology for Human Identification*, volume 6944, Orlando, FL, USA, 2008.

[47] M. Freire-Santos, J. Fierrez-Aguilar, and J. Ortega-Garcia. Cryptographic Key Generation Using Handwritten Signature. In *Proceedings of SPIE Conference on Biometric Technologies for Human Identification*, volume 6202, pages 225–231, Orlando, USA, April 2006.

[48] W. K. French. Automatic recognition of fingerprints by sensing the skin surface with electrical apparatus. International Business Machine Corporation, New York, US Patent No. 3,231,861, 1966.

[49] B. Fu, S. X. Yang, J. Li, and D. Hu. Multibiometric cryptosystem: Model structure and performance analysis. *IEEE Transactions on Information Forensics and Security*, 4(4):867–882, December 2009.

[50] J. Galbally, C. McCool, J. Fierrez, S. Marcel, and J. Ortega-Garcia. On the vulnerability of face verification systems to hill-climbing attacks. *Pattern Recognition*, 43(3):1027–1038, 2010.

[51] F Galton. Personal identification and description. *Nature*, 38:201–202, 1888.

[52] F. Gray. Pulse code communication. US Patent No. 2,632,058, 1953.

[53] J. I. Hall. Notes on coding theory. `http://www.mth.msu.edu/~jhall/classes/codenotes/GRS.pdf`, 2001.

[54] F. Hao, R. Anderson, and J. Daugman. Combining Crypto with Biometrics Effectively. *IEEE Transactions on Computers*, 55(9):1081–1088, September 2006.

[55] W Herschel. Skin furrows of the hand. *Nature*, 23:76, 1880.

[56] W. J. Herschel. *The Origin of Finger-Printing*. Humphrey Milford, Oxford University Press, 1916.

[57] C. Hill. Risk of masquerade arising from the storage of biometrics. Masters thesis, Australian National University, 2001.

[58] R. Hill. *A First Course In Coding Theory*. Oxford University Press, 1988. p. 102.

[59] A. K. Hrechak and J. A. Mchugh. Automated fingerprint recognition using structural matching. *Pattern Recognition*, 23(8):893–904, 1990.

[60] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[61] A. K. Jain and J. Feng. Latent Palmprint Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1032–1047, June 2009.

[62] A. K. Jain, L. Hong, and R. Bolle. On-line Fingerprint Verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):302–314, April 1997.

[63] A. K. Jain and D. Zongker. Feature Selection: Evaluation, Application, and Small Sample Performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.

[64] A.K. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP Journal on Advances in Signal Processing*, 2008:1–17, 2008.

[65] T. Y. Jea and V. Govindaraju. A minutia-based partial fingerprint recognition system. *Pattern Recognition*, 38(10):1672–1684, 2005.

[66] H.-K. Jee, S.-U. Jung, and J.-H. Yoo. Liveness Detection for Embedded Face Recognition System. *International Journal of Biomedical Sciences*, 1(4):235–238, 2006.

[67] X. Jiang and W.-Y. Yau. Fingerprint minutiae matching based on the local and global structures. In *Proc. 15th International Conf. Pattern Recognition*, volume 2, pages 1038–1041, 2000.

[68] A. Juels and M. Sudan. A Fuzzy Vault Scheme. In *Proceedings of IEEE International Symposium on Information Theory*, page 408, Lausanne, Switzerland, 2002.

[69] A. Juels and M. Wattenberg. A Fuzzy Commitment Scheme. In *Proceedings of Sixth ACM Conference on Computer and Communications Security*, pages 28–36, Singapore, November 1999.

[70] E.J.C. Kelkboom, X. Zhou, J. Breebaart, R.N.J. Veldhuis, and C. Busch. Multi-algorithm fusion with template protection. In *Proc. IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems*, Washington, DC, September 2009.

[71] T. A. M. Kevenaar, G. J. Schrijen, M. vanderVeen, A. H. M. Akkermans, and F. Zuo. Face recognition with renewable and privacy preserving binary templates. In *Proc. AutoID*, pages 21–26, 2005.

[72] K. Kollreider, H. Fronthaler, and J. Bigun. Evaluating Liveness by Face Images and the Structure Tensor. In *Proceedings of Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, pages 75–80, Buffalo, USA, October 2005.

[73] C. Lee, J. Y. Choi, K. A. Toh, and S. Lee. Alignment-Free Cancelable Fingerprint Templates Based on Local Minutiae Information. *IEEE Trans. Systems, Man, and Cybernetics, Part B*, 37(4):980–992, 2007.

[74] E. C. Lee, K. R. Park, and J. Kim. Fake Iris Detection by Using Purkinje Image. In *Proceedings of International Conference on Biometrics*, volume LNCS 3832, pages 397–403, Hong Kong, China, 2006.

[75] Y. J. Lee, K. Bae, S. J. Lee, K. R. Park, and J. Kim. Biometric Key Binding: Fuzzy Vault based on Iris Images. In *Proceedings of Second International Conference on Biometrics*, pages 800–808, Seoul, South Korea, August 2007.

[76] J. Li, Y. Wang, T. Tan, and A.K. Jain. Live Face Detection Based on the Analysis of Fourier Spectra. In *Proceedings of SPIE Conference on Biometric Technology for Human Identification*, volume 5404, pages 296–303, Orlando, USA, March 2004.

[77] J.-P. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *in Proc. 4th Int. Conf. Audio- And Video-Based Biometric Person Authentication*, page 393402, 2003.

[78] L. Ma, T. Tan, y. Wang, and D. Zhang. Personal identification based on iris texture analysis. *IEEE Trans. on PAMI*, 25(12):15191533, December 2003.

[79] D. Maio, D. Maltoni, J. L. Wayman, and A. K. Jain. FVC2002: Second Fingerprint Verification Competition. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 811–814, Quebec City, Canada, August 2002.

[80] E. Maiorana and P. Campisi. Fuzzy commitment for function based signature template protection. *IEEE Signal Processing Letters*, 17(3):249–252, 2010.

[81] 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*, 32:1643–1651, 2011.

[82] P. Mohanty, S. Sarkar, and R. Kasturi. From scores to face templates: a model based approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(12):2065–2078, 2007.

[83] E. Mordini and S. Massari. Body, biometrics and identity. *Bioethics*, 22(9):488–498, 2008.

[84] Morpho. Morpho metamatcher. `http://www.morpho.com/morphotrak/MorphoTrak/mt_multi-biometrics.html`.

[85] Morpho. Morphokit sdk. `http://www.morpho.com/identification/secure-biometric-access/software-development-kit/`.

[86] D. Muramatsu. Online signature verification algorithm using hill-climbing method. In *Proc. IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, pages 133–138, 2008.

[87] A. Nagar and A. K. Jain. On the Security of Non-Invertible Fingerprint Template Transforms. In *Proceedings of IEEE Workshop on Information Forensics and Security*, London, UK, December 2009.

[88] A. Nagar, S. Rane, and A. Vetro. Privacy and Security of Features extracted from Minutiae Aggregates. In *Proceedings IEEE Intl Conf. on Acoustics, Speech and Signal Processing*, pages 524–531, Dallas, TX, March 2010.

[89] K. Nandakumar. Fingerprint matching based on minutiae phase spectrum. In *Proc. International Conference on Biometrics*, New Delhi, India, 2012.

[90] K. Nandakumar and A. K. Jain. Multibiometric template security using fuzzy vault. In *Proc. Biometrics: Theory, Applications and Systems*, 2008.

[91] K. Nandakumar, A. K. Jain, and S. Pankanti. Fingerprint-based Fuzzy Vault: Implementation and Performance. *IEEE Transactions on Information Forensics and Security*, 2(4):744–757, December 2007.

[92] K. Nandakumar, A. Nagar, and A. K. Jain. Hardening Fingerprint Fuzzy Vault Using Password. In *Proceedings of Second International Conference on Biometrics*, pages 927–937, Seoul, South Korea, August 2007.

[93] K. Niinuma, U. Park, and A. K. Jain. Soft biometric traits for continuous user authentication. *IEEE Transactions on Information Forensics and Security*, 5(4):771–780, 2010.

[94] K. A. Nixon and R. K. Rowe. Multispectral Fingerprint Imaging for Spoof Detection. In *Proceedings of SPIE Conference on Biometric Technology for Human Identification*, volume 5779, pages 214–225, Orlando, USA, March 2005.

[95] Nurotechnology. Verifinger SDK 4.2. http://www.neurotechnology.com.

[96] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1965.

[97] S. Parthasaradhi, R. Derakhshani, L. A. Hornak, and S. A. C. Schuckers. Time-Series Detection of Perspiration as a Liveness Test in Fingerprint Devices. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 35(3):335–343, 2005.

[98] H. C. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[99] A. M. Polinsky and D. L. Rubinfeld. A model of optimal fines for repeat offenders. *Journal of Public Economics*, 46(3):291–306, December 1991. `http://www.nber.org/papers/w3739.pdf`.

[100] M. Potzsch, T. Maurer, L. Wisk ott, and C. von-der Malsburg. Reconstruction from graphs labeled with responses of gabor filters. In *Proc. International Conference of Artificial Neural Networks*, pages 845–850, Bochum, 1996.

[101] Federal Bureau of Investigation. Integrated Automated Fingerprint Identification System. Available at `http://www.fbi.gov/hq/cjisd/iafis.htm`.

[102] Unique Identification Authority of India. Multipurpose National Identity Card. Available at `http://uidai.gov.in/`.

[103] P. Pudil, J. Novovicova, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125, 1994.

[104] F. Quan, S. Fei, C. Anni, and Z. Feifei. Cracking cancelable fingerprint template of Ratha. In *International Symposium on Computer Science and Computational Technology.*, volume 2, pages 572–575, 2008.

[105] N. K. Ratha, R. M. Bolle, V. D. Pandit, and V. Vaish. Robust fingerprint authentication using local structural similarity. In *Proc. Fifth IEEE Workshop on Applications of Computer Vision*, pages 29–34, 2000.

[106] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle. Generating Cancelable Fingerprint Templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):561–572, April 2007.

[107] D.L.T. Rhodes. Methods for Binary Multidimensional Scaling. *Neural Computation*, 14:1195–1232, 2002.

[108] R. N. Rodrigues, L. L. Ling, and V. Govindaraju. Robustness of multimodal biometric fusion methods against spoof attacks. *Journal of Visual Languages and Computing*, 20(3):169 – 179, 2009.

[109] A. Ross, K. Nandakumar, and A. K. Jain. *Handbook of Multibiometrics*. Springer, 2006.

[110] A. K. Ross, J. Shah, and A. K. Jain. From Template to Image: Reconstructing Fingerprints From Minutiae Points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):544–560, 2007.

[111] M. Savvides and B. V. K. Vijaya Kumar. Cancellable Biometric Filters for Face Recognition. In *Proceedings of IEEE International Conference Pattern Recognition*, volume 3, pages 922–925, Cambridge, UK, August 2004.

[112] W. J. Scheirer and T. E. Boult. Cracking fuzzy vaults and biometric encryption. In *Proceedings of Biometrics Symposium*, Baltimore, USA, september 2007.

[113] R. Seacord. *Secure Coding in C and C++*. Addison-Wesley, 2005.

[114] D. R. Setlak. Fingerprint Sensor Having Spoof Reduction Features and Related Methods. US Patent No. 5,953,441, 1999.

[115] S. Shah. Enhanced iris recognition: Algorithms for segmentation, matching and synthesis. Master's thesis, Department of Computer Science and Electrical Engineering, West Virginia University, 2006.

[116] S. W. Shin, M-K Lee, D. Moon, and K. Moon. Dictionary attack on functional transform-based cancelable fingerprint templates. *ETRI Journal*, 31(5):628–630, 2009.

[117] C. Soutar. Biometric system performance and security. In *Proc. Auto ID*, New Jersey, 1999.

[118] A. Stoianov. Cryptographically secure biometrics. In *Proceedings of SPIE*, volume 7667 (1), page 76670C, 2010.

[119] Y. Sutcu, Q. Li, and N. Memon. Secure Biometric Templates from Fingerprint-Face Features. In *Proceedings of CVPR Workshop on Biometrics*, Minneapolis, June 2007.

[120] Y. Sutcu, S. Rane, J. Yedidia, S. Draper, and A. Vetro. Feature extraction for a slepian-wolf biometric system using ldpc codes. In *Proceedings of the IEEE International Symposium on Information Theory*, Toronto, Canada, July 2008.

[121] Y. Sutcu, H. T. Sencar, and N. Memon. A Secure Biometric Authentication Scheme Based on Robust Hashing. In *Proceedings of ACM Multimedia and Security Workshop*, pages 111–116, New York, USA, August 2005.

[122] A. B. J. Teoh, A. Goh, and D. C. L. Ngo. Random Multispace Quantization as an Analytic Mechanism for BioHashing of Biometric and Random Identity Inputs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1892–1901, December 2006.

[123] A. B. J. Teoh, D. C. L. Ngo, and A. Goh. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recognition*, 37(11):2245–2255, 2004.

[124] A. B. J. Teoh, K.-A. Toh, and W. K. Yip. 2^N Discretisation of BioPhasor in Cancellable Biometrics. In *Proceedings of Second International Conference on Biometrics*, pages 435–444, Seoul, South Korea, August 2007.

[125] V. Testoni and D. Kirovski. On the inversion of biometric templates by an example. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 1830 – 1833, 2010.

[126] M. Tico and P. Kuosmanen. Fingerprint matching using an orientation-based minutia descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):1009–1014, 2003.

[127] S. Tulyakov, F. Farooq, P. Mansukhani, and V. Govindaraju. Symmetric hash functions for secure fingerprint biometric systems. *Pattern Recognition Letters*, 28(16):2427–2436, 2007.

[128] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive NeuroScience*, 3(1):71–86, 1991.

[129] U. Uludag and A.K. Jain. Attacks on Biometric Systems: A Case Study in Fingerprints. In *Proceedings of SPIE Conference on Security, Seganography and Watermarking of Multimedia Contents VI*, pages 622–633, San Jose, USA, January 2004.

[130] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. V. Jawahar. Blind authentication: a secure crypto-biometric verification protocol. *IEEE Transactions on Information Forensics and Security*, 5(2):255–268, 2010.

[131] C. Vielhauer, R. Steinmetz, and A. Mayerhofer. Biometric Hash Based on Statistical Features of Online Signatures. In *Proceedings of* 16^{th} *International Conference on Pattern Recognition*, volume 1, pages 123–126, Quebec, Canada, August 2002.

[132] P. R. Vizcaya and L. A. Gerhardt. A nonlinear orientation model for global description of fingerprints. *Pattern Recognition*, 29(7):1221–1231, 1996.

[133] A. Wahab, S.H. Chin, and E.C. Tan. Novel approach to automated fingerprint recognition. *Proc. IEEE Visual Image Signal Processing*, 145(3):160–166, 198.

[134] W. Wertelecki and C.C. Plato. *Dermatoglyphics - Fifty Years Later*. Alan R. Liss, Inc., 1979.

[135] J.D. Woodward. Biometrics: privacy's foe or privacy's friend? *Proceedings of the IEEE*, 85(9):1480 – 1492, 1997.

[136] H. Xu, R.N.J. Veldhuis, T.A.M. Kevenaar, A.H.M. Akkermans, and A.M. Bazen. Spectral minutiae: A fixed-length representation of a minutiae set. In *Proceedings of IEEE Computer Vision and Pattern Recognition. Workshop on Biometrics*, Anchorage, Alaska, 2008.

[137] Y. Yamazaki, A. Nakashima, K. Tasaka, and N. Komatsu. A study on vulnerability in on-line writer verification system. In *Proc. International Conference on Document Analysis and Recognition (ICDAR)*, pages 640–644, 2005.

[138] B. Yang and C. Busch. Parameterized geometric alignment for minutiae-based fingerprint template protection. In *Proc. IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems*, Crystal City, Washington DC, September 2009.

[139] S. Yang and I. Verbauwhede. Automatic Secure Fingerprint Verification System Based on Fuzzy Vault Scheme. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 609–612, Philadelphia, USA, March 2005.

[140] S. Yoon, J. Feng, and A. K. Jain. Altered fingerprints: Analysis and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):451–464, March 2012.

[141] Y. Zhu, S. C. Dass, and A. K. Jain. Statistical Models for Assessing the Individuality of Fingerprints. *IEEE Transactions on Information Forensics and Security*, 2(3):391–401, September 2007.

[142] J. Zuo, N. K Ratha, and J. H Connell. Cancelable iris biometric. In *Proceedings of the 19th International IAPR Conference on Pattern Recognition (ICPR 2008)*, pages 1–4, 2008.