Privacy Protecting Biometric Authentication Systems

by

Alisher Kholmatov

Submitted to the
Faculty of Engineering and Natural Sciences
in partial fulfillment of the requirements
for the degree of
DOCTOR OF PHILOSOPHY

Sabanci University
January, 2008

Privacy Protecting Biometric Authentication Systems

APPROVED BY

Assoc. Prof. Berrin Yanıkoğlu (Thesis Supervisor)	.....
Prof. Aytül Erçil	.....
Prof. Lale Akarun	.....
Assoc. Prof. Albert Levi	.....
Assoc. Prof. Erkay Savaş	.....

DATE OF APPROVAL: 17.01.2008

January, 2008

to my beloved wife Zulfiya, our daughter Maryam and her future brothers & sisters

Acknowledgments

My sincerest thanks go to Prof. Berrin Yanıkoğlu for all her support and patience in assisting me through out the course of my Ph.D. studies. I appreciate her valuable advice and efforts she offered. It has been a great honor for me to work under her guidance.

I would also like to thank all my jury members, Prof. Aytül Erçil, Prof. Lale Akarun, Prof. Albert Levi and Prof. Erkay Savaş for their equally valuable support generously given during the writing of my thesis. I am grateful to Prof. Özgür Gürbüz, Prof. Ibrahim Tekin and Prof. Hakan Erdoğan for their valuable advice and discussions.

Special thanks go to my colleagues and friends Mustafa Parlak, Yasser Elkahlout, Ilknur Durgar, Özlem Çetinoğlu and many others. I appreciate their friendship and sympathetic help which made my life easier and more pleasant during my studies.

Lastly, I would like to thank my parents and my wife Zulfiya for their enormous encouragement, assistance and patience, for without them, this work would not have been possible.

ABSTRACT

Privacy Protecting Biometric Authentication Systems

As biometrics gains popularity and proliferates into the daily life, there is an increased concern over the loss of privacy and potential misuse of biometric data held in central repositories. The major concerns are about i) the use of biometrics to track people, ii) non-revocability of biometrics (eg. if a fingerprint is compromised it can not be canceled or reissued), and iii) disclosure of sensitive information such as race, gender and health problems which may be revealed by biometric traits. The straightforward suggestion of keeping the biometric data in a user owned token (eg. smart cards) does not completely solve the problem, since malicious users can claim that their token is broken to avoid biometric verification altogether. Put together, these concerns brought the need for privacy preserving biometric authentication methods in the recent years.

In this dissertation, we survey existing privacy preserving biometric systems and implement and analyze fuzzy vault in particular; we propose a new privacy preserving approach; and we study the discriminative capability of online signatures as it relates to the success of using online signatures in the available privacy preserving biometric verification systems. Our privacy preserving authentication scheme combines multiple biometric traits to obtain a multi-biometric template that hides the constituent biometrics and allows the possibility of creating non-unique identifiers for a person, such that linking separate template databases is impossible. We provide two separate realizations of the framework: one uses two separate fingerprints of the same individual to obtain a combined biometric template, while the other one combines a fingerprint with a vocal pass-phrase. We show that both realizations of the framework are successful in verifying a person's identity given both biometric

traits, while preserving privacy (i.e. biometric data is protected and the combined identifier can not be used to track people).

The *Fuzzy Vault* emerged as a promising construct which can be used in protecting biometric templates. It combines biometrics and cryptography in order to get the benefits of both fields; while biometrics provides non-repudiation and convenience, cryptography guarantees privacy and adjustable levels of security. On the other hand, the fuzzy vault is a general construct for unordered data, and as such, it is not straightforward how it can be used with different biometric traits. In the scope of this thesis, we demonstrate realizations of the fuzzy vault using fingerprints and online signatures such that authentication can be done while biometric templates are protected. We then demonstrate how to use the fuzzy vault for *secret sharing*, using biometrics. Secret sharing schemes are cryptographic constructs where a secret is split into shares and distributed amongst the participants in such a way that it is reconstructed/revealed only when a necessary number of share holders come together (e.g. in joint bank accounts). The revealed secret can then be used for encryption or authentication. Finally, we implemented how correlation attacks can be used to unlock the vault; showing that further measures are needed to protect the fuzzy vault against such attacks.

The discriminative capability of a biometric modality is based on its uniqueness/entropy and is an important factor in choosing a biometric for a large-scale deployment or a cryptographic application. We present an individuality model for online signatures in order to substantiate their applicability in biometric authentication. In order to build our model, we adopt the Fourier domain representation of the signature and propose a matching algorithm. The signature individuality is measured as the probability of a coincidental match between two arbitrary signatures, where model parameters are estimated using a large signature database. Based on this preliminary model and estimated parameters, we conclude that an average online signature provides a high level of security for authentication purposes.

Finally, we provide a public online signature database along with associated testing protocols that can be used for testing signature verification systems.

Özet

Kişisel Gizliliği Sağlayan Biyometrik Doğrulama Sistemleri

Biyometrik sistemlere rağbetin artması ve günlük hayatımızın bir parçası haline gelmeleriyle birlikte, bu tür sistemlerde kişisel gizlilik ihlali ile ilgili olan endişelerin de arttığını gözlemlemekteyiz. Özellikle merkezi veritabanlarında saklanan biyometrik verilerin amaç dışı kullanılabiliyor olması kaygıları iyice körüklemektedir. Biyometrik verilerle ilgili ana endişeleri şu şekilde özetlemek mümkündür: i) kişileri takip etme amaçlı kullanılmaları, ii) geri dönüşümlerinin olmaması (örn. kopyalanan/çalınan parmak izlerinin değiştirilemiyor olması), iii) ırk, cinsiyet ve sağlık durumu gibi hassas bilgileri ifşa edebiliyor olmaları. Hemen akla gelen öneri, biyometrik verilerinin kişinin sahip olduğu aygıtlarda saklanması (örn. akıllı kart), problemi tam olarak çözemez, çünkü kötü niyetli kullanıcılar aygıtlarının bozulduğunu veya çalındığını iddia edip biyometrik doğrulamayı tamamen devre dışı bırakabilirler. Bahsi geçen endişeler ve sorunlar birleştiğinde, kişisel gizliliği sağlayan biyometrik doğrulama yöntemlerine duyulan ihtiyaç önemli ölçüde artmaktadır.

Tezimizin ana araştırma katkılarını şu şekilde özetleyebiliriz: kişisel gizliliği sağlayan biyometrik sistemlerinin irdelenmesi, önemli birisinin gerçeklenmesi ve analizi; çoklu biyometrik verileri birleştirerek kişisel gizliliği sağlayan yeni bir yöntemin önerilmesi; dinamik imzaların var olan kişisel gizliliği sağlayan yöntemler çerçevesinde kullanılabilirliğini saptamak amacıyla, ayırt edicilik kapasitelerinin araştırılması. Önerdiğimiz çoklu biyometrik yöntemi, birden çok biyometrik veriyi birleştirerek bu bilgilerin gizliliğini sağlar. Ayrıca çoklu biyometrik şablonlarının bulunduğu bir veritabanı, tek bir biyometrik (örn. parmak izi) kullanılarak izinsiz sorgulanamaz. Gizlilik unsurlarını sağlamasına ek olarak bu yöntemin ayrıca kimlik doğrulamada da tekli bir sisteme göre daha başarılı olduğunu deneysel sonuçlarımızla

kanıtlamaktayız. Tez kapsamında yöntemimizin iki ayrı gerçeklemesini göstermekteyiz: birinde aynı kişinin iki farklı parmak izini diğerinde ise parmak izini ve sesli şifresini birleştirebildiğimizi ve kişi doğrulamada başarıyla kullanılabildiklerini göstermekteyiz.

Bulanık Kasa adı verilen yöntem, biyometrik bilgilerin gizlenmesinde kullanılabilecek bir yöntem olarak ön plana çıkmıştır, ancak değişik biyometrik verilerinin bulanık kasa çerçevesinde nasıl kullanılacakları konusunda açıklık yoktur. Tezimiz kapsamında, bulanık kasa yöntemini parmak izi ve dinamik imzalar ile gerçekledik, ayrıca sır paylaşımında nasıl kullanılabileceğini gösterdik. Kriptografide oldukça yaygın olan sır paylaşım yöntemleri, gizli kalması gereken bilginin, sadece birkaç kişinin bir araya gelmesiyle açığa çıkması gereken durumlarda kullanılır. Bulanık kasa yöntemi ile geliştirdiğimiz sistemde, ancak belirlenen sayıda kişilerin parmak izlerinin bir araya gelmesi ile açığa çıkarılan sır, hem doğrulama hem de şifreleme amaçlı kullanılabilmektedir. Son olarak da, tezimiz kapsamında bulanık kasa yönteminin ilinti saldırılarına karşı dayanıksız kalacağı iddiasını test ettik; bu kapsamda, önerilen saldırıları gerçekleyip, deneysel olarak sıklıkla başarılı olduklarını gösterdik.

Bir biyometrik verinin ayırt edicilik kapasitesi onun bireyselliğine dayanmaktadır ve verinin büyük ölçekli ya da kriptografik uygulamalarda tercih edilmesinde önemli bir etkendir. Tezimiz kapsamında, dinamik imzaların doğrulama amaçlı kullanılabilirliğini desteklemek amacıyla, ortalama bir imzanın sahip olduğu tahmin edilme olasılığını modelledik. Bunun için imzaların Fourier katsayılarına dayanan bir gösterim ve özgün eşleştirme yöntemi önerdik ve bunları kullanarak iki imza arasındaki rastlantısal eşleşme olasılığını hesapladık. Önerilen modele ve kestirilen değişkenlere dayanarak, dinamik imzaların oldukça düşük (10^{-4}) bir tahmin edilme olasılığı olduğu sonucuna varmaktayız.

Son olarak da tez kapsamında toplanan dinamik imzaları, kapsamlı test protokolleri ile birleştirerek araştırma amaçlı kullanıma açtık.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

With demanding security regulations throughout the world and increasing amount of valuable services provided using the Internet and other networked media, the assurance of secure and privacy preserving identity authentication became a crucial issue. Assurance of both security and privacy is itself a very challenging task since security requirements are prone to undermine a user's privacy. While private information (eg. social security number, marital status, facial photo etc.) collected during enrollment for a particular service increases security, unauthorized disclosure of such information undermines the prerogative of privacy. Likewise, a person's actions can be tracked by linking different sources of information and utilizing that person's uniquely identifying surrogates (eg. credit card and social security numbers, fingerprints, etc.). In this chapter, we elaborate on commonly utilized user authentication methods; we overview general aspects of biometrics and discuss its associated privacy concerns.

There are three major identity authentication approaches: knowledge-based, token-based and biometrics [1]. Knowledge-based methods rely on information that only a genuine user is supposed to know, such as passwords or PINs. Token-based authentication requires that the user presents a legitimate token which is provided by a recognized authority. Commonly used tokens are smart cards with built-in micro chips which can store a user's personal information, access rights, etc. Biometric authentication requires that a subject possesses a body trait (such as a fingerprint or iris pattern) or is able to reproduce a particular behavioral task (such as a signature

or spoken password) that matches the previously stored template, in order to be positively verified.

Password and token-based authentication methods have noticeable shortcomings which we shortly discuss. An ordinary person may have difficulties with remembering a password which is complex enough to be guessed by someone else. As a result, people commonly write down their passwords on unprotected media (eg. piece of paper, back of a credit card, etc.) or use passwords associated with themselves [2] (eg. birthdays, telephone numbers, names of the relatives, nicknames of pets, etc.) which enable attackers perform brute force attacks based on social engineering. Furthermore, in order to reduce number of passwords required to remember, people tend to use the same password or a small set of passwords for different applications [3]. Hence if a password is revealed by compromising one of the applications, the attacker gets an access to all other applications used by that user. Resetting a user‘s password is not a cheap procedure either, as it may seem; according to a password survey conducted on corporate employees, the cost for resetting a password is estimated as 30-50$ dollars. On the other hand, token-based methods have their own disadvantages as smart cards or other tokens can be broken, lost or stolen. Finally, passwords and tokens are not tightly coupled with their owner's identity, thus can not provide non-repudiation (not being able to deny involvement). Biometrics emerged as the technology promising to alleviate these shortcomings. It provides convenience such that there is no need to remember or carry anything, user simply has it as a part of his/her body. Biometric traits can not be shared, copied, lost or stolen thus provide non-repudiation.

A generic biometric authentication system consists of two main parts: enrollment and verification. During the enrollment, a user is asked to submit his/her biometric trait, which is captured and digitized by a biometric sensor. Discriminative feature values are then extracted and stored in the form of a template in the system's database, along with the user's identity. To authenticate him/herself, a subject submits his/her biometric trait (query) which is then compared against the template corresponding to the claimed identity. Depending on the dissimilarity between the

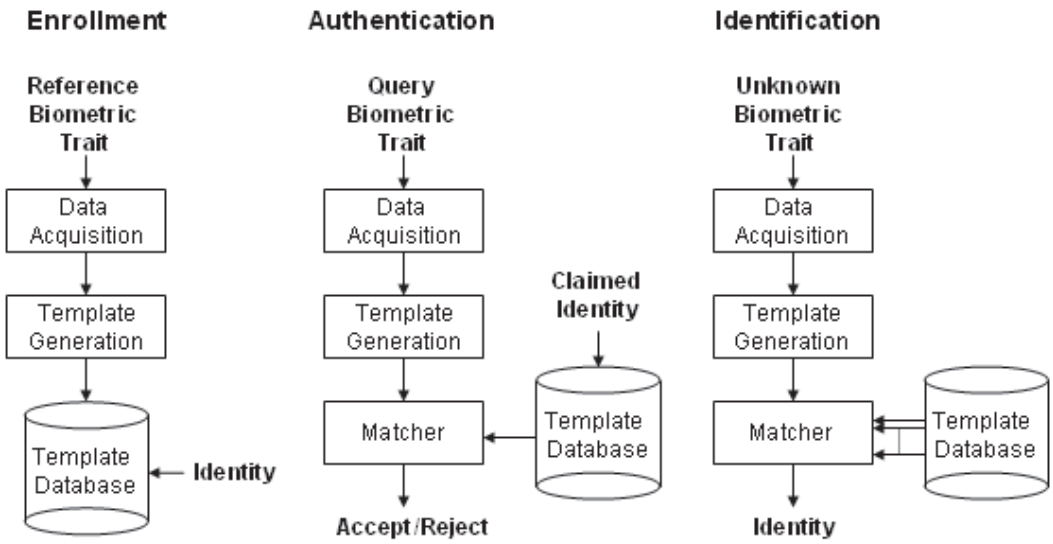

Figure 1.1: Main blocks of biometrics based user enrollment (left), authentication (middle) and identification (right).

query and the template, the system either rejects or accepts the user as forgery or genuine, respectively. Figure 1.1 schematically depicts biometric enrollment and authentication phases (leftmost and middle columns).

Biometric data can also be used for identification, which is the task of searching the database for the most similar biometric trait(s), given a biometric trait with an unknown identity. For example, when a police finds an unknown fingerprint in a crime scene, they search their records in order to find if it corresponds to a person in their database. Identification is a much more time consuming operation than authentication, as it requires a large number of comparisons. Figure 1.1 (rightmost column) schematically depicts the identification task.

In evaluating the performance of a biometric verification system, there are two important factors: false rejection rate (FRR) of genuine traits and false acceptance rate (FAR) of impostor traits. Since these two error rates are inversely related, a commonly reported performance measure is the Receiver Operating Characteristic (ROC) curve which shows how true accept rate (1-FRR) changes with FAR, for different acceptance thresholds. When only a single performance measure is required,

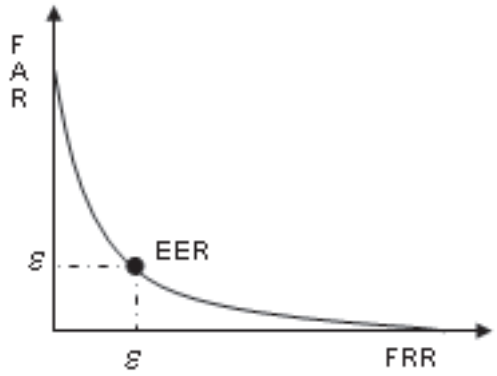

Figure 1.2: A sample error trade-off curve.

for instance while comparing different systems, the equal error rate (EER) that denotes the point on the ROC curve where FAR equals FRR, is often reported. The Figure 1.2 illustrates above mentioned concepts.

Proper biometric traits must be selected for a particular security application. The biometric chosen to be used in a military application may be different than the one used for access control for an apartment building. Biometric traits can be classified according to different criteria, such as existence, permanence, uniqueness, ease of measurement, difficulty of being copied or reproduced, acceptance by the general public, and cost of deployment. Table 1.1 represents an informal categorization of some of the widely used biometric traits, which is intended to give the rough picture. As can be seen, there are tradeoffs between these criteria. Often, a biometric which is unique and difficult to measure and forge (e.g. retina), is also less acceptable by the public and has higher deployment costs.

The discriminative capability of a biometric is based on its uniqueness/entropy across the population which can be measured as the probability of a coincidental match between the biometric data of two different subjects. For example, the uniqueness of fingerprints determines the probability of correspondence between two arbitrary selected fingerprints. Assessing the entropy of a biometric trait is not as

	Uniqueness	*Acceptance*	*Hard To Forge*	*Permanence*
Retina	High	Low	High	High
Iris	High	Medium	High	High
Fingerprint	High	Medium	Medium	Medium
Face	Medium	High	Medium	Low
Hand	Medium	High	Medium	Medium
Signature	Medium	High	Medium	Medium
Voice	Low	High	Low	Low

Table 1.1: Relative categorization of biometric traits.

straightforward as it is with passwords (i.e. by calculating all available passwords), since simply calculating the entropy of a biometric signal without regard to the intra-class variations would result in an unrealistically optimistic entropy measure. Instead, the entropy of a biometric trait is established either by a theoretical model and/or by a large scale empirical assessment. A biometric trait can be classified as strong or weak according to its uniqueness degree. For instance, iris, fingerprint and retina are considered strong while voice and gait are not. We broadly discuss on these matters in Chapter 5.

Strong biometrics can be used to identify the owners, which rises certain privacy concerns. Although, privacy has broad aspects and its boundaries may differ from society to society, in our study we consider privacy as the ability of individuals to control the flow of information about themselves and reveal such information selectively with or without passing the right to disclose it to third parties. The major concerns associated with biometrics are about i) the use of biometrics to track people, ii) non-revocability of certain biometric traits once compromised, and iii) disclosure of sensitive information such as race, gender and health problems, which may be revealed by some biometric traits.

Tracking of individuals can be performed by linking separate databases which have records or transactions associated with biometric traits of a person and revealing where and when the person has been, what he/she has purchased etc. The

parallel can be made to credit cards that have unique identification numbers. Once a person makes a purchase, a transaction is being recorded into his/her bank's database. Such transactions record where and when the purchase is made, along with the amount and other essential information. So if the credit card transaction database is shared with other institution(s) that has a link between the credit card number and the identity information of its owner, then it is a straightforward to track that person's whereabouts, shopping attitudes etc.

Additionally, tracking can be performed without sharing any such database. Most of the biometric traits can be easily acquired without notice and special involvement of their owners. For instance, facial and iris images can be easily photographed using a digital camera posed apart enough not to be noticeable by a subject. Likewise, people generally leave their fingerprints on whatever they touch and registering someone's voice without being noticed is relatively easy. Once obtained and registered, their owners can be tracked. Most importantly, once a biometric is stolen, it is stolen forever, no revocation or replacement is generally possible, except for some of the behavioral biometrics such as signature.

Another privacy concern issue is about the fact that certain biometric data may reveal sensitive information such as race, gender and health problems [4]. For instance, according to the study of McLean [5] the diseases causing fragility of palm skin and nails can disclose certain genetic disorders. Chen [6] mentions that abnormalities of fingerprint ridges may be caused either by certain chromosomal disorders, which are associated with Down, Turners and Klinefelters syndromes, or by nonchromosomal disorders that may be due to leukemia, breast cancer and Rubella syndromes. Similarly, Schuster [7] identified a correlation between the so-called digital-arc fingerprint pattern and chronic intestinal pseudo-obstruction disease, conjectured to be caused by a genetic disorder. The retina and iris biometrics may reveal diabetes, arteriosclerosis and hypertension as well as their own diseases [8]. Hence, if a biometric is used to find out about such sensitive information which may be later used to deny health insurance, employment or any such privilege, it is surely a privacy breach. On the other hand, although biometric traits may reveal certain

diseases, we don't know whether biometric templates themselves (e.g. fingerprint minutiae) can disclose any such sensitive information.

Yet another privacy concern is the function creep: initially, biometric traits may be used solely for important authentication purposes, but their use may become so common place in the future with potentially unforeseeable consequences. Social Security Number (SSN) practiced in United States is a good example for such concerns. SSN was initially used in record keeping of Social Security taxes. Later, the Internal Revenue Service (IRS) started using the SSN for tax identification purposes and currently SSN is required for employment, insurance, driving licence and many more [4].

Some straightforward privacy preserving solutions can come to mind: i) instead of using central databases, smart card like tokens can be used to store biometric templates, ii) biometric templates can be stored in an encrypted form rather than being stored as a plain feature vector. However, none of these solutions is actually practical for preserving privacy. In particular, forgers can claim that their card is broken or stolen and avoid biometric verification altogether. Besides, restoration of broken or lost tokens may require referring to a central database for certain legitimacy verification. Encrypting biometric templates will alleviate certain privacy issues that arise with unintended sharing of the databases. In such situation, linking databases without encryption keys will be infeasible. However, this requires management of encryption keys, which has its own privacy concerns and additional security challenges.

Tomko [9] proposed to use biometric traits only as encryption keys without storing biometric templates. In an example solution, a user's fingerprint would be used to encrypt a secret information required to access different applications/services. Since the secret information is encrypted and the access to different applications is supposed to be using different secret information, linking databases to track people across applications will be infeasible. Although this is a good solution, there are drawbacks associated with it. In particular, extracting a cryptographic key from a noisy and variable data such as biometrics is a very challenging task and remains

an open research area.

In this thesis, we review state-of-the art research on privacy protection in biometric systems (Chapter 2) and propose our own privacy preserving framework with its practical realizations using fingerprint and voice biometrics (Chapter 3). We demonstrate how online signatures can be used for cryptographic key generation and how biometric traits can be used for secret sharing (Chapter 4). Then, in order to substantiate the use of online signatures in authentication and cryptographic key generation, we present a theoretical model measuring the discriminative capability of online signatures (Chapter 5). Finally, we present an online signature database along with associated testing protocols, to be used in testing online signature verification systems (Chapter 6).

Chapter 2

Previous Work

In this chapter, we review previously proposed methods which are applicable for privacy protection in biometric authentication systems. We review biometric cryptosystems which utilize both biometric traits and cryptographic protocols to achieve higher security and user convenience (Section 2.1). For the sake of completeness, we also review privacy enhancing methodologies that prevent using biometric identification in surveillance video records (Section 2.2).

2.1 Template Protection and Biometric Cryptosystems

Biometric systems are gaining popularity as more trustable alternatives to password-based security systems, since there are no passwords to remember and biometrics cannot be stolen and are difficult to copy. Biometrics also provide non-repudiation (an authenticated user cannot deny having done so) because of the difficulty in copying or stealing one's biometrics. On the other hand, biometric measurements are also known to be variable and noisy; the same biometric trait of a person may slightly vary between consecutive acquisitions due to the noise in the acquisition process, surrounding environment, injury, or even a bad mood. For example, different impressions of a fingerprint can greatly vary due to differences in the dryness of the finger tip, the levels and location of pressure applied to the finger tip, or different sensors, as demonstrated in the Figure 2.1.

Figure 2.1: Different impressions of the same fingerprint, demonstrating distortion and noise introduced during the acquisition process.

Biometric template refers to the information extracted from a biometric and stored as the reference. For instance, if a fingerprint is used, the biometric template may consists of features extracted from the fingerprint image (e.g. minutiae points indicating the branching and ending points of the ridges of the fingerprint). Biometric template protection, in turn, generally refers to protecting one's biometric data or biometric template from unauthorized access or unintended use (e.g. to track the person or to gather sensitive information about the person). As mentioned in the previous chapter, biometric template protection is especially important because biometrics cannot be revoked and re-issued once compromised.

Uludag et al. makes the distinction between two general approaches within what they call crypto-biometric systems, according to the coupling level of cryptography and biometrics [10]: *Biometrics-based key release* refers to the use of biometric authentication to release a previously stored cryptographic key. Biometric authentication is used as a wrapper, adding convenience to traditional cryptography where the user would have been in charge of remembering his/her key; however the two techniques are only loosely coupled. *Biometrics-based key generation* refers to extracting/generating a cryptographic key from a biometric template or construct. In this case, biometrics and cryptography are tightly coupled: the secret key is bound to the biometric information and the biometric template is not stored in plain form.

In its most basic sense, generating a cryptographic key from a biometric template (say fingerprints) has not been very successful, as it involves obtaining an *exact* key from a highly variable data.

Soutar et al. [11] proposed a method to bind cryptographic keys with the image of the fingerprint. The key is released only upon the presentation of the genuine fingerprint's image and can be used for user authentication and additionally for cryptographic encryption/decryption operations. If a key is somehow compromised a new one can be generated and re-associated with the fingerprint image by re-enrolling a user. The algorithm is based on the correlation filter function which is calculated from reference fingerprint images. The filter function, when applied onto the genuine fingerprint image, is supposed to produce consistent output pattern. The method also make use of error correction codes to account for small variations in the filter output. Main drawbacks of the Soutar et al.'s work are: i) the formal and systematic cryptographic security analysis of the method is not provided [12,13] and ii) method requires aligned fingerprints (reference and query fingerprint images must be aligned precisely) which brings user inconvenience i.e. each time users must place their fingerprints on a sensor almost the same way.

Teoh et al. proposed to map a biometric feature vector onto a randomly generated orthonormal vector space in order to obtain a revocable binary representation of a biometric, which is then used for authentication [14, 15]. We shortly describe here an implementation using fingerprints [14] while the other implementation using face biometric [15] is very similar. In order to extract fingerprint feature vector, an integrated Wavelet and Fourier-Mellin transform [16] is applied to a fingerprint image. Then, a number of orthonormal vector spaces are generated by applying Gram-Schmidt transform to a randomly generated matrices. The generation of random matrices is controlled by a seed used to initialize a random number generator. That seed is then stored to a user's token (eg. smart card). A number of generated matrices corresponds to the number of bits desired to represent the fingerprint (best results are reported for 60 and 80 bits). Inner products between the feature vector and each of the orthonormal vector spaces are calculated. The results of

inner products are binarized and concatenated into a bit string which is stored in the system database. During verification, user's bit string is similarly calculated using the query fingerprint and the seed stored on his/her token. The user is successfully authenticated if the Hamming distance between the calculated bit string and that stored on the system's database is small. Authors report 0% ERR using fingerprint representation of 40 and more bits. One of the drawbacks is that the method requires robust detection of fingerprint's core point around which the image is cropped. The other drawback is the requirement of secure storage media such as smart card for a random number generator's seed, which reduces convenience of the proposed method.

Davida et al. [17, 18] and Hao et al. [19] proposed the use the IrisCode, a 2048 bit string extracted from iris texture proposed by Daugman [20], to generate cryptographic keys. We review only the work of Hao et al. as it provides more practical implementation and contains less restrictive assumptions compared to that of Davida et al. Daugman has shown that genuine IrisCode's may have up to 30% bit difference due to noise and image processing artifacts [20], thus they can not be directly applied for encryption. In order to obtain a reliable iris representation, Hao et al. analyzed the reasons behind the differences and devised a 2-stage error correction algorithm which is based on Hadamard and Reed-Solomon error correction codes [21, 22]. The key is bind to and retrieved from the IrisCode using some helper data which must be stored on a secured media (authors assume that it is stored on the smart card). Possession of both a genuine iris image and the helper data is required in order to successfully release the associated key. The key can be revoked by changing the helper data. Authors report that they could generate 140-bit keys at 0.47% FRR and 0% FAR. Main drawback of the scheme is that it requires secured media to store helper data which reduces convenience of the method.

Monrose et al. [23] propose a method to enhance security of a conventional password based authentication system using keystroke behaviors of its users. The security of the method is based on the difficulty of the polynomial reconstruction problem. For each user a $m \times 2$ (row x column) table containing evaluation pairs

(i.e. [x,P(x)]) of a $m-1$ degree polynomial (P) is created. Initially, each cell contains valid evaluation pair (i.e. one ling on the polynomial), but as the user logs into the system, his/her consistent keystroke features are being estimated and cells, identified according to these features, are being perturbed such that corresponding evaluation pair is no more ling on the polynomial. When a user logs into the system, his/her keystroke features are being calculated and the evaluation pairs corresponding to these features are used to reconstruct the polynomial. If the polynomial is correctly reconstructed, the user is successfully authenticated. It is assumed that even if the attacker will intercept the password, he/she will not be able to reproduce keystroke dynamics of the genuine user, thus will fail to correctly identify the valid evaluation pairs and reconstruct the polynomial. Authors were able to increase the security/entropy of passwords by approximately 15 bits, which is indeed not very substantial. Additionally, Monrose et al. demonstrate extension of their method to the voice biometric, where they succeed in obtaining a 60-bit cryptographic keys from the uttered pass phrases [24, 25]. However, even a 60-bit cryptographic keys are considered week for the most of the contemporary cryptographic applications.

Recent work of Juels et al. [13] is also classified as biometrics-based key generation, allowing for a tight coupling of cryptography and biometrics. Juels and Wattenberg proposed the fuzzy commitment scheme [26]; later Juels and Sudan extended it to the *fuzzy vault* scheme [13] and described how it can be used to construct/release an encryption key using one's biometrics: a secret (cryptographic key) is *locked* using a biometric data of a person, such that someone who possesses a substantial amount of the locking elements (e.g. another reading of the same biometric) would be able to decrypt the secret [13]. The fuzzy vault scheme is classified as a key-generation scheme in Uludag et al., because of its tight coupling of cryptography and biometrics [10]. However, in the sense that the biometric data releases a previously stored key, it can also be seen as a releasing mechanism. Clancy et al. [27], Yang and Verbauwhede [28] and Uludag et al. [29] implemented the fuzzy vault using fingerprints, making simplifying assumptions about the biometric data. We describe details of the fuzzy vault scheme as well as provide our own implementations using

fingerprints and online signatures in the Chapter 4.

Feng and Wah proposed a private key generation method using online signatures [30]. The method is based on feature quantization and used only dynamic features of a signature. First, the range of each feature is calculated across all subjects to obtain database boundaries for that feature. During enrollment, user boundaries are found similarly and the database range for each feature is divided into bins of size equal to the user's range. Then, the indices of the bins where the user's features are mapped, are concatenated into a single vector from which the cryptographic hash value is calculated. In other words, quantization is done adaptively for each user. The hash value is then used to calculate a private key for that user. Authors report a performance of 8% equal error rate in generating the keys. They also analyze the entropy of each feature and conclude that online signatures contain on average 40 bits of entropy, calculated as the sum of individual feature entropies. Since the features may not be independent, this estimate of the signature entropy is an overestimate.

Ratha et al. suggest [31] and implements [32] a framework of cancelable biometrics, where a biometric data undergoes a predefined non-invertible distortion during both enrollment and verification phases; if the transformed biometric is compromised, the user is re-enrolled to the system using a new transformation. Likewise, different applications are also expected to use different transformations for the same user. Although this framework hides original (undistorted) biometric and enables revocation of a (transformed) biometric, it introduces the management of transform databases, and still requires registration of reference points.

Tuyls et al. demonstrated a practical application of their previously proposed privacy protecting theoretical scheme [33,34] to the ear canal biometric [35]. A fixed length feature vector is extracted from a headphone to ear canal transfer function [36], which is then used to encode a secret key. After selecting an appropriate encoding function, each dimension of the feature space is quantized into a fixed number of bins. During encoding, a helper data is generated, which contains offsets used in mapping the test biometric's feature values to their corresponding bins.

The helper data and a cryptographic hash value of the secret key are stored in the systems database.

During authentication, the query biometric's feature values are summed with the corresponding helper data offsets, and the resulting values are mapped on to the bins. Depending on whether a feature value is mapped to an even (0) or odd (1) indexed bin, its corresponding bit value is generated. Finally, a hash value of the generated bit string is compared to that stored in the system's database. It is assumed that a few bit errors can be fixed, prior to calculation of the hash value, using an appropriate error correction code. In their theoretical work, authors provide systematic proofs that the proposed method doesn't leak information sufficient to guess the key or reveal the biometric template. On the other hand, the proposed method requires that the template and query biometric data are precisely aligned as well as the intra-class variation and the noise introduced during the data acquisition can be handled by proper feature space quantization. Another drawback is that the maximum bit size of the secret key is limited by the number of extracted biometric features.

2.2 Privacy Protection in Surveillance Video

Privacy preserving in surveillance video is also a very important and widely concerning issue, as people can be identified and tracked across different video recordings using biometric identification technology such as face or gait recognition.

Governments and private sector are spending considerable portions of their budgets for surveillance. For instance, according to Tyler [37] Britain has approximately 4.2 million of Closed Circuit TV (CCTV) cameras installed. It is estimated that an ordinary British citizen might be captured by more than 300 separate cameras on an average day [37]. In such circumstances, if recordings of these cameras were accessible to unintended authorities, then revealing where and for how long the person has been, whom s/he has met, what s/he has bought or where s/he has ate can be accomplished by identifying faces, gaits or voices of recorded people, if the video

quality allowed such identification.

Last but not least, video recordings are kept for a long time and can be redistributed very quickly and to a large audience. For example, a video clip, containing private life events of a person, can be relatively easy broadcast using the Internet, which indeed occurs frequently. Even if the clip is removed from the web site shortly after, it is impossible to destroy all of the copies already downloaded by its viewers. Thus the clip can appear at a later time and continue to reveal someone's private life forever.

Privacy issues associated with video surveillance are being raised by many institutions and individuals [4, 38–40]. However, engineering solutions that preserve privacy must be also developed. Privacy protection in surveillance video is rather new and emerging research area. In this section, we review a few of the available approaches aiming for privacy protection in surveillance video.

Masking the eyes or the complete face of an individual with a black bar and changing his/her voice during various TV programs (e.g. secret agent talking about successful operation) can be considered as initial attempts to preserve privacy in video records. However, while preserving privacy of people recorded on the video, such methods are of limited interest since these can not be used as evidence for prosecution. It is worth to mention that saving two copies of a video (i.e. one with all private regions masked and the original copy encrypted) does not solve the problem, as it requires additional investments for storage and enhancements/enforcements to maintain the overall security and integrity of the entire system.

A similar approach is proposed by Newton et al. [41], where authors argue that masking faces is of limited interest for various multimedia applications. Instead, they propose to de-identify (i.e. degrade) facial features such that face recognition software will be unable to correctly identify degraded faces. While preserving privacy, this approach has similar drawbacks with the aforementioned method.

Sony Inc. proposed and patented a method to detect skin regions and replace them with arbitrary colors, which to some extent prevents determination of the race [42]. It is clear that such precaution is also of limited interest for privacy

protection as face identification is still possible. Likewise, racial origin can still be estimated based on other facial features (eg. structure of the eyes, skulls or lips).

Senior et al. proposed a privacy preserving video console [43], which is rather a framework for managing video content of the surveillance video using computer vision techniques and cryptography. The system records the video in an encrypted form and re-renders demanded video portion or provides just a particular event according to the user's privileges. Implementation of this system and/or applying it to existing systems are the main challenges.

Boult [44] proposed to obscure the private content of an image/video using invertible cryptographic transform. The region containing the private information is cropped from the image or the video frame just after a lossy encoding operation (eg. DCT, DWT). Then, that region is encrypted using any arbitrary encryption technique (eg. DES, AES), and mapped back to the image for final encoding. Since the encryption transforms the given data to a complete random stream, the cropped region is completely obscured, which enhances privacy. Figure 2.2 demonstrates such masking. Only authorities possessing encryption key (presumably law enforcement authorities) can decrypt the obscured regions and reveal the identities of the corresponding individuals. Boult implemented this technique to only JPEG images, and claims that the compression overhead introduced by his approach will not exceed 10% if implemented for MPEG video.

Dufaux and Ebrahimi proposed a region-based transform-domain scrambling technique [45, 46]. Firstly, regions of private information (eg. faces or complete body) are detected on a video frame by means of computer vision techniques. These regions are then scrambled (i.e. obscured) by flipping signs of the corresponding coding transform coefficients (eg. DCT or DWT) during the encoding. The flipping is controlled by a secret key and is invertible, meaning that someone who possesses the key can reconstruct the original images/frames. Additionally, regions of arbitrary shapes can be scrambled and the degree of the obscuration is adjustable through the number of flipped coefficients.

To enhance privacy, Zhang et al. [47] proposed a method to replace sensitive

Figure 2.2: Image regions containing faces are cropped, then encrypted and mapped back to their original places for privacy protection.

regions of a video record with their corresponding backgrounds and store removed regions as a watermark in the corresponding video. When required, authorities possessing the encryption key can reveal the watermark and reconstruct the original video footage. Additionally, a digital signature is embedded into the video header to detect any tampering. The main drawback of the proposed method is that it highly increases the frame rate.

Providing quantitative measure for the privacy enhancement is another research area. Jonathon Phillips [48] studied the inverse relation between privacy and surveillance performance. He proposed a privacy operating characteristic curve (POC), which is an analogy of receiver operating curve (ROC), which is commonly used to assess false accept rate versus false reject rate of a biometric verification system. Using POC, system administrators can select an appropriate operating point for a surveillance system with regard to a privacy enhancing level. The POC curve is obtained by degrading sensitive information content in a corresponding video record, which corresponds to a certain privacy level, and measuring its corresponding surveillance performance at that level.

Chapter 3

Multi-Biometric Templates for Privacy Protection

We propose a biometric authentication framework which is based on the idea of using multiple biometric traits to increase both privacy and security of the verification system. Specifically, we combine different biometric traits of an individual to create a multi-biometric template. Due to the difficulty of separating the multi-biometric template into its constituents, the individual biometrics are protected. Also, if one uses separate sets of biometrics for different security applications, the resulting multi-biometric templates are different, preventing tracking by linking several databases. Security is also increased since verification requires each component biometrics. As a particular example, we demonstrate a fingerprint verification system that uses two separate fingerprints of the same individual. A multi-biometric template is created by overlaying the minutiae points of two fingerprints and then storing the combination in the central database.

3.1 Overview of Fingerprint Verification

Fingerprints have a long history of being used for person identification. Although different fingerprint representations are available, the minutiae point representation is by far the most prevailing and popular [49]. Minutiae points of a fingerprint are the landmark points formed by the ridge structure of the corresponding fingerprint. Figure 3.1 demonstrates different minutia point types on a sample fingerprint image. Relative ridge structure of fingerprints and their minutiae points are established

before birth and are accepted to be unique to each individual. Even identical twins have different fingerprints, due to the fact that the formation of each fingerprint is dependent not only on the individual's DNA, but is also highly effected by the micro-environment (pressure and temperature differences, flow of fluids, etc.) surrounding the fingerprint tip [50].

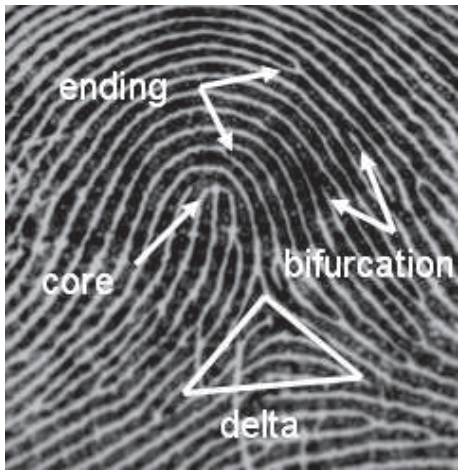

Figure 3.1: Most commonly used fingerprint minutiae points: delta, core, ridge ending and ridge bifurcation.

There are several methods proposed in the literature for automatic minutiae extraction [51,52]. Majority of such methods extract minutiae from a skeletonized (all ridge lines are reduced to 1-pixel thickness) fingerprint ridge pattern. Prior to detection, the fingerprint image is adaptively enhanced, making use of the overall ridge flow, then binarized and finally thinned. Figure 3.2 illustrates minutiae extraction process. The detection may result in spurious or missing minutiae, which is due to the skin cuts and imperfections or noise introduced during the fingerprint image acquisition. In order to purify the detected minutiae, a post-processing is generally performed [53, 54].

Two fingerprints are accepted as similar if there is a sufficient number of matching minutiae. The acceptance threshold differs from country to country; for instance,

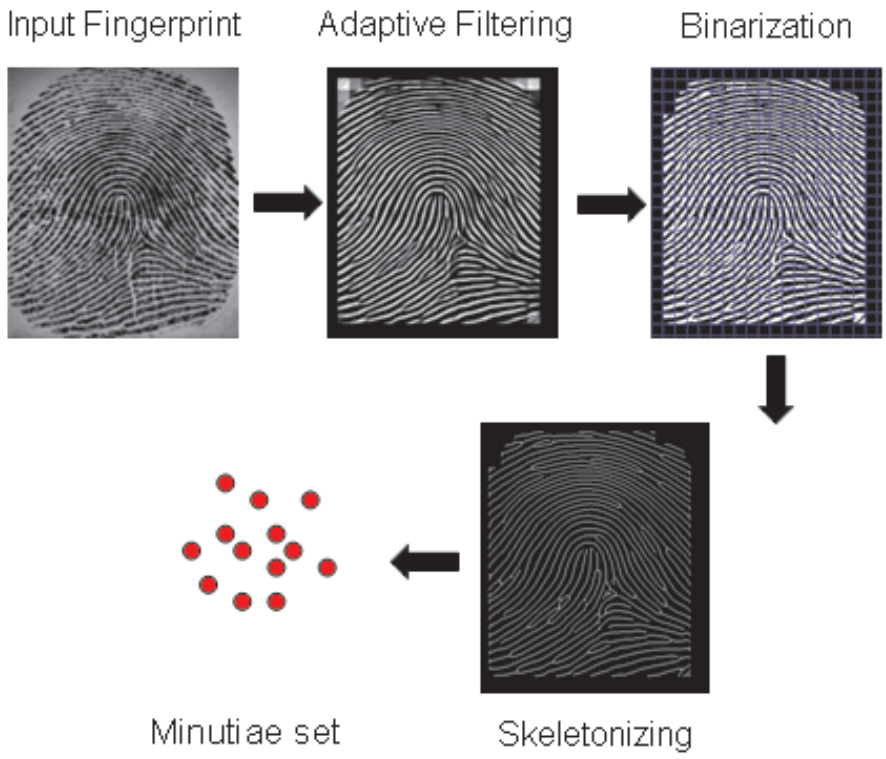

Figure 3.2: Illustration of the commonly implemented minutiae extraction method.

USA's F.B.I. require 12, British Scotland Yard 16 and Interpol 12 minutiae point correspondence [55]. Jain et al. proposed an automatic fingerprint matching algorithm where the ridge information is used to align the corresponding minutiae sets and small displacements between matching minutiae are handled by accepting a match if it is within a bounding box [49, 56]. Ratha et al. proposed a matching technique based on graph representation, which is constructed for both the query and template fingerprints [57]. The state-of-the-art performance of automatic fingerprint verification algorithms varies between 0.01-2.15% depending on the difficulty of the database used for testing. The above mentioned performance results are reported by internationally accepted fingerprint verification competitions [58–61] and the fingerprint vendor technology evaluations [62].

Multiple biometric modalities are used to increase the security of the system or to address cases where a user may not posses a required biometric (eg. due to injury).

For example, a user may be asked to put his fingerprint and pronounce a codeword in order to be positively authenticated. The combination/fusion of different biometric data can occur at various levels, namely decision, score or feature levels. For feature level fusion, features extracted from different biometric traits can be combined for a single classifier. In the case of decision level fusion, separate classifiers can operate independently on different biometric traits and their matching scores are combined for the final decision [49,63,64]. Several different systems are proposed for combining multiple biometrics; for instance voice and face biometrics [65–67] and fingerprint and face biometrics [68].

3.2 Multi-Biometric Authentication Framework

In this section, we formalize and demonstrate our framework using fingerprints. We also explain how the proposed framework can be extended using the voice biometric.

3.2.1 Feature Extraction

We used ridge ending and ridge bifurcation minutiae points as our features, since these are the most commonly utilized fingerprint features. We only use minutiae point locations, discarding the additional information associated with the minutiae points (eg. ridge orientation, grayscale neighborhood) as it may leak sensitive information.

Since the aim of this work is to conceptually demonstrate the framework, we used manually labeled minutiae locations, to avoid errors that may be caused by an imperfect minutiae extraction module. Hence the features extracted from one fingerprint image is a set of minutiae locations (x,y).

3.2.2 Multi-Biometric Template Generation

In order to create a multi-biometric template, a user submits the impressions of his/her two different fingerprints, hereafter denoted by A and B. Minutiae point

locations of these two fingerprints are detected and then scrambled with each other to hide their source. Here we introduce a scrambling operator (denoted by $\cup$), which offsets one minutiae set with respect to the other set, roughly aligning their centers of gravity. This combined minutiae set ($A \cup B$), which constitutes the multi-biometric template, is then stored in the system database.

Figure 3.3: Two fingerprints A (on the left) and B (in the middle) are combined to form the multi-biometric template ($A \cup B$ on the right). Minutiae points are differently marked for the sake of clarity.

The template creation process is illustrated in the Figure 3.3, where the combined minutiae set is shown on the right. Note that in this multi-fingerprint template, minutiae origins (i.e. their corresponding fingerprints) are illustrated with separate markers solely for the clarity of the illustration; in reality, they are indistinguishable in the multi-biometric template.

Note that the template can be generated by many different fingerprint pairs; as such, it is *not* a unique identifier of the person. Likewise, two different persons can engage in creating a shared multi-biometric template. For instance, such shared templates can be created for an application requiring presence of two authorizing people in order to approve or initiate a particular task.

3.2.3 Matching

When a person is to be authenticated, he/she again submits new impressions of his/her two fingerprints (hereafter denoted by A' and B'), *both* of which are used to verify his/her identity. The verification consists of two sequential steps: in each step a single query fingerprint is matched against the template of the claimed identity.

In the first step, A' is matched against the multi-biometric template and all matching points are discarded from the template, resulting in $A \cup B - A'$. We introduced a *fuzzy* set subtraction operator (indicated by $-$) that allows for some tolerance in matching. Then, the second fingerprint B' is matched against the remaining minutiae points in the template. In both of the cases, the matching is done by finding the correspondence between the minutiae points of a query fingerprint and the multi-biometric template. Both the minutiae extraction and the point correspondence algorithm are non-essential to the proposed method and any previously developed minutiae detection or correspondence algorithms can be used. The matching process for a case where both of the query fingerprints are genuine, is illustrated in the Figure 3.4. Note that even though the minutiae points are marked in the figures with circles and square, indicating their corresponding source fingers, that is done solely for the clarity and the sake of explanation. As we previously mentioned, origins of minutiae points are not kept in the template.

Finally, we calculate a matching score using the Jaccard index between the two sets involved in the last matching; in other words, the percentage of matching points in B' and the remainder set:

$$Jaccard(A \cup B - A', B') = 2 \times \frac{\left|(A \cup B - A') \cap B'\right|}{\left|(A \cup B - A') \cup B'\right|} \tag{3.1}$$

Here we introduce a *fuzzy* set intersection operator (indicated by $\cap$) which tolerates for some misalignment between corresponding minutiae points; and $|X|$ indicates the cardinality of the set X. The person is authenticated if the match score is above a threshold, which is selected in this case as the point that corresponds to the equal error rate.

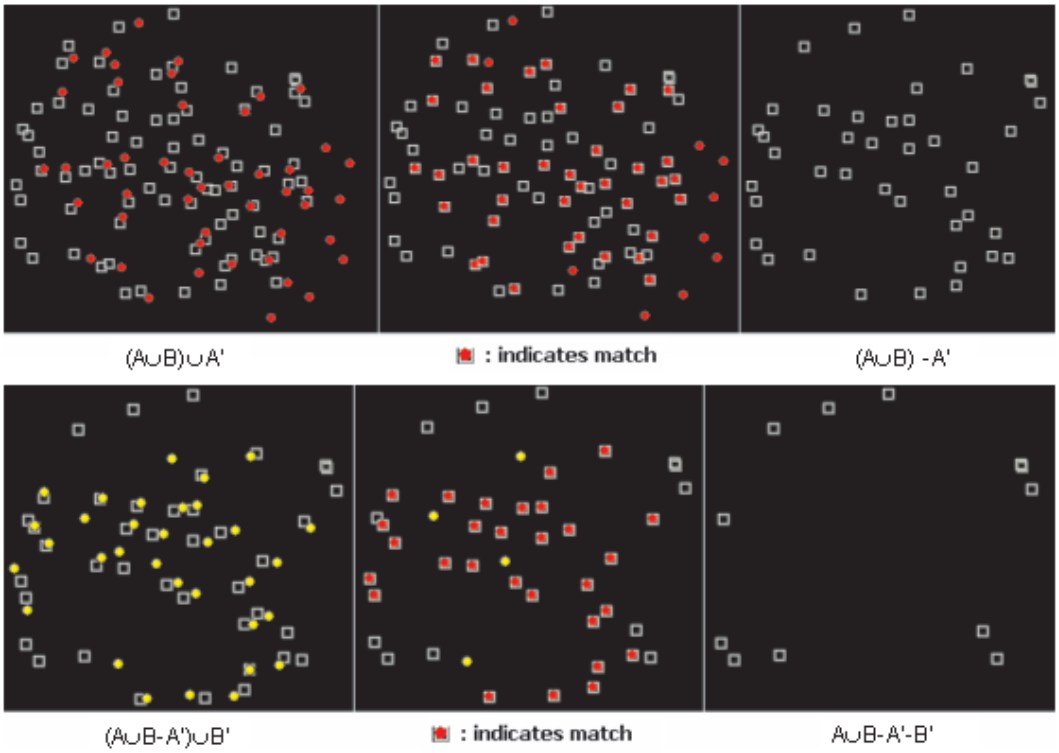

Figure 3.4: An illustration of matching two genuine fingerprints (A' and B') against the multi-biometric template.

Note that even though the overall match score seems to be based solely on B''s match, if A' was not successfully matched, it would be reflected in the final score since many minutiae points would be left unmatched, making the denominator large. There is still a bit of asymmetry since unmatched points of A' are not factored in the matching score. This could be remedied by reversing the order of the match sequence (first B' and then A') and averaging the two resulting scores.

We consider three different cases in order to show how the proposed scheme works. In the first case, both of the query biometrics are genuine: A' will match $A \cup B$, leaving mostly points of B and the rest is equivalent to the verification with a single biometric. In case A' matches A perfectly and B' matches B perfectly, the resulting score is 1. The second case assumes that A' is forgery while B' is genuine: A' will still have a good match to $A \cup B$ which has a large number of points (roughly twice as many than A'). But then, even though B' is a genuine biometric, it will not

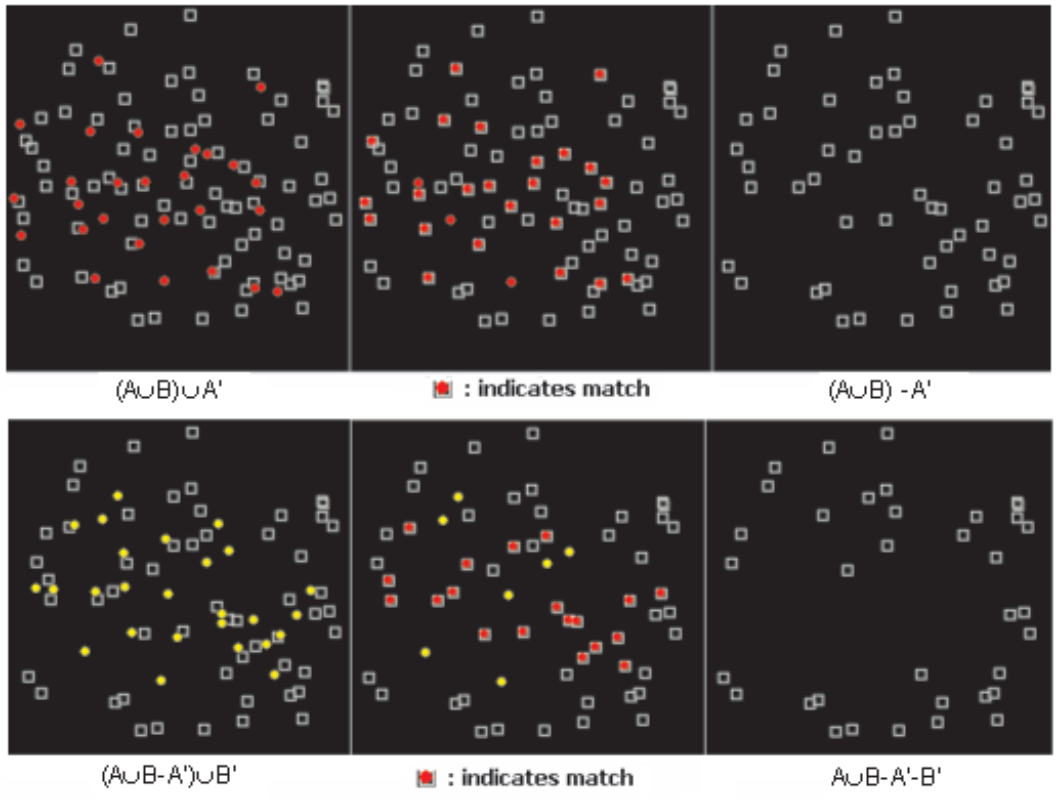

Figure 3.5: An illustration of matching a forgery (A') and a genuine (B') fingerprint against the multi-biometric template.

have a good match with $(A \cup B - A')$. Figure 3.5 shows a sample for this case. The third case is where A' is genuine and B' is forgery: A' will have a good match to $(A \cup B)$ leaving mostly the B component, so the rest is equivalent to the verification of a single forgery biometric, which will not result in a good match.

The number of matching minutiae obtained in the first step is significantly higher than if two corresponding fingerprints (A and A') were matched, due to the large number of minutiae points in the multi-biometric template (about twice as many minutiae points as a single fingerprint). In particular, fingerprints with few minutiae points match to several multi-biometric templates with large sets of minutiae points. However, this does not reduce the effectiveness of the system: if any minutiae from B are matched by A', it will reduce the match score *only* if it matters (if A's and B's minutiae are nearby, it does not matter whose minutiae are matched). On

the other hand, this property makes it very difficult to search the multi-biometric database using only a single fingerprint to find matching records. Note that this is the intended result, since it prevents unauthorized uses of the database, for instance performing identification with single fingerprints from another database or from a crime scene. Hence, not only that the individual fingerprints are hidden (by way of having two sets of points scrambled together), but also searching a multi-biometric database is impractical, as explained in the experimental results section.

3.2.4 Experiments

A total of four fingerprint impressions (two from one finger and two from another finger) are collected from each of the 100 people contributing to the database. One impression from each finger (A and B) is added to the reference set: they are used to form the multi-biometric for the person. The remaining two fingerprint impressions (A' and B') are added to the test set: they are used to authenticate the person. Figure 3.6 shows a quadruple from the database: the top row is the reference set and the bottom row is the test set. Notice that the fingerprints have some missed minutiae, due to the shifts and deformations introduced during the acquisition of the imprints.

Once the data is collected, a multi-biometric template is constructed from the reference set of each person in the database. For testing, we used the test set of a person and did the sequential matching. Both of these steps are detailed in the previous section. The minutiae points are marked manually, but the matching is done automatically. Notice that the manual labeling of the minutiae points is not essential: any reasonably successful minutiae detection and matching algorithm can be applied. The automatic matching is done via an exhaustive matching algorithm that aligns two point sets by finding the best alignment over all translations and rotations, allowing for some elastic deformation of the fingerprint (accepting two points as matching if they are within a small threshold in this alignment). Since the aim of this work is to introduce the idea of a multi-biometric templates, we only focused on showing that the resulting multi-biometric preserves privacy, while still

Figure 3.6: Sample quadruple fingerprints from the database. Top row shows fingerprints A and B; bottom row shows fingerprints A' and B', left to right.

successfully authenticating a person. Hence, minutiae detection and matching were assumed to be given or were simply implemented.

Using our database and the proposed method explained in the Section 3.2, we obtained a 2% false reject rate (FRR). In other words, 2 out of 100 people in the database were not authorized using their second set of fingerprints (A' and B'). On the other hand, when each of these fingerprint pairs were used as a forgery for all other people (for a total of 9900=100*99 data points), only 1.8% were falsely accepted (FAR). The equal error rate (EER) was approximately at 1.9%. Most of the errors were due to fingerprints that had significant stretching between two imprints, as these are not well matched using our simple matching algorithm. Our other biggest source of error is due to fingerprints that have missing left or right parts (i.e. fingerprint occlusions), due to pressure being applied to one side of the finger while taking the imprint.

In order to test how much error is introduced with the new authentication scheme

(using two fingerprints instead of one), we calculated the error rates for a biometric system that matched single fingerprints (e.g. A versus A') using the same minutiae detection and matching algorithms. The matching score used was the ratio of the number of matching points over the total number of points in the matched and the reference fingerprints:

$$Jaccard(A, A') = 2 \times \frac{\left|A \cap A'\right|}{\left|A \cup A'\right|} \tag{3.2}$$

In this task, the FRR was found to be 3%: in other words, 6 fingerprints were falsely rejected out of 200 fingerprints (100x2). When each fingerprint was used as forgery for all the others, the FAR was found to be 2%. Hence, the multi-biometric scheme not only did not introduce any additional errors, but rather reduced the error rate. This is in fact as expected and observed in other multi-modal biometric systems, since we are given more identifying information about the person. The acceptance thresholds for both of the previous tests were set on the test set, for both tasks, in order to give the EER. Since FAR and FRR are inversely proportional, this is a common practice and does not introduce undue advantage.

Finally, we performed a privacy analysis in order to assess the degree of privacy the multi-biometric template framework provides. We assessed whether a single fingerprint was sufficient to search the multi-biometric template database (i.e. given only one fingerprint, what are the chances to correctly identify a person?). The scoring method used was based on the proportion of the minutiae points of the presented fingerprint (A') that matched the template set ($A \cup B$):

$$Jaccard((A \cup B), A') = 2 \times \frac{\left|(A \cup B) \cap A'\right|}{\left|(A \cup B) \cup A'\right|} \tag{3.3}$$

Using this score, the fingerprint to be identified matched with the corresponding multi-biometric template (i.e. the template gave the highest match score) for only 24% of the test cases. When considering top-5 results (accepting the person as correctly identified if the correct template was in the top-5 highest matching alternatives), the identification rate rose up to 39%. While 39% may seem a large number, in a larger database, these numbers would be expected to be lower, making

it infeasible to search the database using single fingerprints.

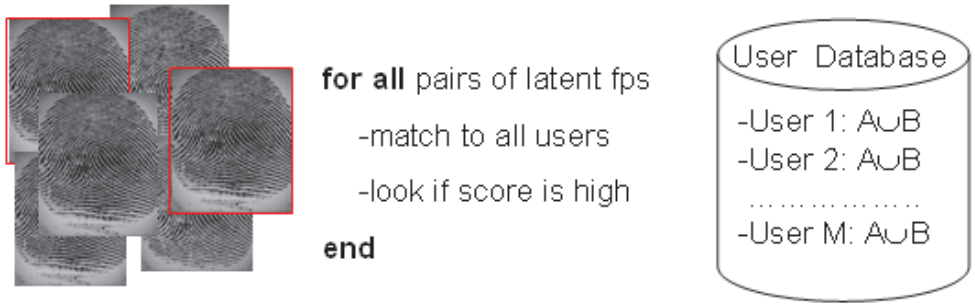

Figure 3.7: An illustration of a multi-biometric database search algorithm using different fingerprint impression combinations.

People easily leave impressions of their fingerprints on surfaces and objects they touch. Given that fact, the natural question would be whether an attacker can search a multi-biometric template database with *combinations* of fingerprints obtained from latent fingerprints? Figure 3.7 demonstrates a pseudocode illustration of such an attack. Generally hundreds of fingerprint impressions are left on the surfaces or objects and their quality is often much worse than regular impressions. Thus, such attacks are infeasible, as a very large number of combinations are needed.

Yet another privacy related question is: Can two multi-biometric template databases be linked together? An attacker may intercept different template databases and try cross-matching their templates. Figure 3.8 illustrates such attack. It will be infeasible if users were to provide different fingerprints pairs for different applications. Also, as we explain in the Section 3.3, this is a natural result when our framework is used with fingerprint and voice modalities.

On the other hand, we have not fully proven that the combined biometric cannot be used to track a person: it may be possible that a certain pattern of minutiae distribution appears only for a given person. However, the addition of minutiae points from the second fingerprint hides these patterns to the largest extent. For future work, one can further look into how to best combine two biometrics (e.g. to

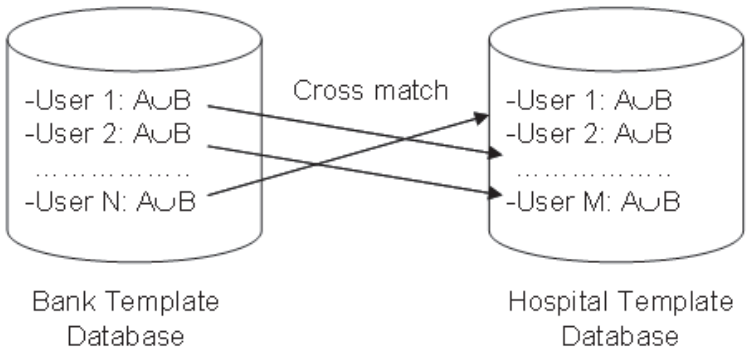

Figure 3.8: An illustration of an algorithm used to cross match and identify corresponding users in two different multi-biometric template databases.

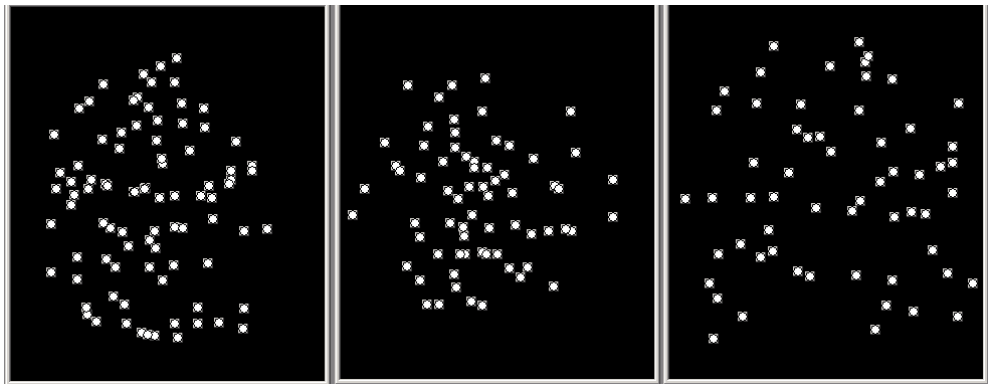

Figure 3.9: Multi-biometric templates created for 3 different people, using 2 of their fingerprints.

disperse the minutiae points as much as possible etc.) so as to hide the most unique features of a fingerprint. Three separate combined fingerprint minutiae are shown in the Figure 3.9 to give some idea of the scrambling that results from the combination of two fingerprints.

3.3 Framework Realization Using Behavioral Traits

One of the substantial drawback with physiological biometric traits is that if they are compromised, their revocation is impossible. On the other hand, changing of a

behavioral trait, such as signature or a vocal password (or pass-phrase), is a relatively easy task; one just needs to make up a new signature or choose another pass-phrase. Additionally, an individual has a freedom of changing and using different instances of his/her behavioral traits for different application (eg. different pass-phrases for different applications), which is not as easy with physiological traits.

In this section, we overview the work of Camlikaya et. al [69] that shows a realization of the multi-biometric template framework, for completeness. In this implementation, fingerprint and voice biometrics are used in the creation of the multi-biometric template, in order to benefit from the aforementioned characteristics. The main challenge in this implementation is the extraction of suitable vocal feature points that can be mapped to the 2D-plane of the minutiae points.

3.3.1 Feature Extraction and Template Generation

The Figure 3.10 demonstrates a typical voice signal. Short spectra of speech signals convey distinguishing information about both the spoken words and the the vocal characteristics of the speaker. Mel Frequency Cepstral Coefficients (MFCC's), which convey both vocal characteristic of a person and uttered pass-phrase, are commonly extracted from voice signal and are further used as feature values. The extraction process of MFCC's is inspired by the human hearing system [70].

Twelve MFCC features are extracted for each phoneme in a user uttered pass-phrase. Then, feature values are binarized according to a threshold decided separately for men and women, and grouped into a 16-bit chunks, each representing one vocal feature point (8 bits for x and 8 bits for y coordinates). There were on average 25 phonemes in each pass-phrase collected for our voice database. This implies that on average 19 vocal feature points ($25 \times 12 \div 16$) are calculated from the voice biometric of a user. The feature extraction is described in detail in Camlikaya et al. [69]. For fingerprints, minutiae points are extracted as described in Section 3.2.1.

In order to be enrolled to the system, the user provides his/her fingerprint and utters a pass-phrase. Extracted fingerprint and vocal minutiae points are merged using our set offset operator($\cup$) to form a multi-biometric template. The process of

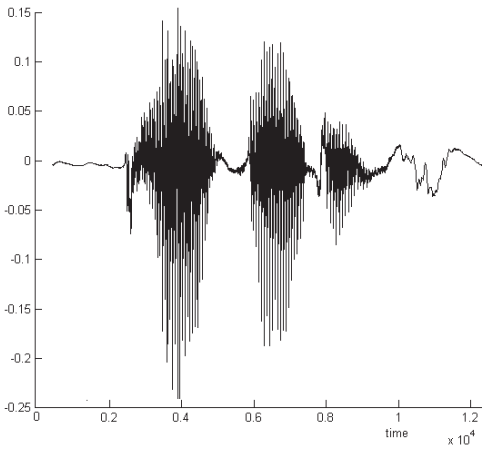

Figure 3.10: A typical digitized voice signal.

the template generation is demonstrated in Figure 3.11.

3.3.2 Matching

During authentication, a subject claims a particular identity ($A \cup B$) and provides his/her fingerprint (A') along with a pass-phrase utterance (B'). The matching of the query fingerprint and utterance are performed in a similar fashion with that of the previous realization of the framework, described in the Section 3.2.3. First, the fingerprint minutiae are matched against the template and then matching points are discarded. Then, the vocal points are matched against remaining template points ($(A \cup B) - A'$). The major difference from the previous matching strategy is that the vocal points are matched to the remaining template points using the Hamming distance, such that the perturbation of each bit has an equal weight. Hence, the coordinates of the remaining template points (i.e. x and y) are concatenated, to reconstruct the corresponding MFCC features and match the vocal points. Decision regarding the authenticity of the user is made based on our previously formulated matching score (the equation 3.1 defined in Section 3.2.3).

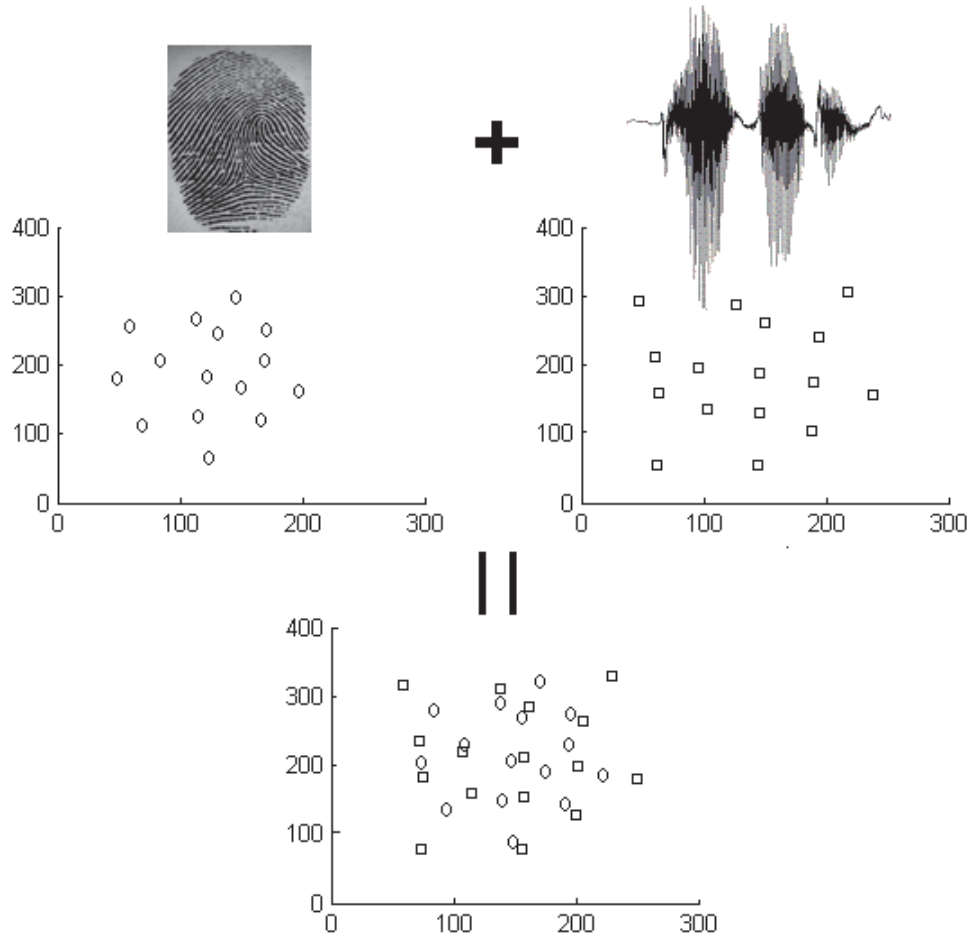

Figure 3.11: A multi-biometric template creation using fingerprint and voice minutiae points.

3.3.3 Experiments

Managing the collection of a multi-modal database is a costly procedure in regards to time and budget; hence, previously collected fingerprint and voice databases were paired to obtain a pseudo-multi-modal database. The fingerprint database is described in the Section 3.2.4. Each of 100 subjects in that database provided 2 impressions of his/her 2 different fingers (400 impressions in total). The voice database consists of 30 other subjects who provided 10 utterances of their pass-phrases as well as attempted to forge someone else's pass-phrase 10 times. Each

pass-phrase is 6-digits long. Voice and fingerprint data were randomly paired as if they belonged to the same person. In this configuration, there are 30 genuine subjects enrolled to the system, where each subject has 2 impressions of his/her fingerprint and 10 utterances of a pass-phrase. All other available fingerprints, as well as 10 forgery utterances are used in forgery attempts for a user's template.

The framework realization is tested using 3 different forgery scenarios: i) attacker uses his/her fingerprint and voice, ii) attacker uses his/her fingerprint and recorded voice of the user, iii) attacker uses collected fingerprint impression of the user and his/her voice to utter user's pass-phrase. The first one is the most common attack scenario where the forger is unable to get genuine biometric traits of the claimed user and uses his/her own or someone else's traits, instead. We achieved 0.77% EER when considering forgeries of this type. A unimodal voice verification method of Camlikaya et al. achieved 3.3% EER [71], using the same feature extraction algorithm and test dataset, which indicates that the multi-biometric template scheme significantly improved results of the unimodal system. For the second and third scenarios, we achieved 5.50% and 21.30% EER. Last results indicate that the fingerprint is relatively more important than voice in the context of this implementation, which is probably due to the relatively simple mapping of voice features to vocal points.

3.4 Summary and Conclusion

With demanding security regulations throughout the world and the large number of valuable services provided using the Internet and other networked media, the assurance of secure and privacy preserving identity authentication became a very crucial issue. In that regard, we have introduced a new concept for combining multiple biometric traits to protect privacy. Our framework combines multiple biometric modalities of a person in order to hide the individual biometrics and create a non-unique multi-biometric template/identifier. We have empirically demonstrated that such multi-biometric identifiers can be successfully used for the authentication,

while searching or linking with other similarly generated identifiers is infeasible, thus privacy preserving. We have successfully demonstrated the applicability of our framework to physiological and behavioral biometric traits, namely fingerprint and voice.

In the framework implementation where minutiae points extracted from two fingerprints and merged to create a multi-biometric template, we achieved 1.9% of equal error rate, which is comparable to the state of the art fingerprint verification systems (although with a smaller database). Additionally, these templates are resilient against identification and tracking, as indicated by a privacy analysis performed on our system. For instance the success of searching correct person within a database of 200 users using a single genuine fingerprint resulted in only a 24%.

The extensibility of our framework to behavioral traits is demonstrated by Camlikaya et al. [69]. In that work, the fingerprint minutiae were scrambled with MFCC based feature points extracted from a vocal pass-phrase. Experimental results showed that the scrambled point set (i.e. minutiae and voice points) is very successful in authentication, while successfully hiding the user's unique biometric features. For this implementation, if a multi-biometric template is somehow compromised, a new one can be regenerated by simply using another pass-phrase.

Chapter 4

Fuzzy Vault for Privacy Protection

Juels and Sudan proposed a scheme called the *fuzzy vault*, which they call an error tolerant encryption operation [13]. The fuzzy vault scheme provides a framework to encrypt ("lock") some secret value (eg. cryptographic key) using an *unordered* set of locking elements as a key, such that someone who possesses a substantial amount of the locking elements will be able to decrypt the secret. Security of the scheme is based on the difficulty of the polynomial reconstruction problem.

In the context of this thesis, we elaborate on the fuzzy vault scheme and its implications on privacy issues. We utilize the fuzzy vault scheme in order to protect online signatures from unintended access and screening. We also show how fuzzy vault can be practically applied for biometric secret sharing. Finally, it was claimed that fuzzy vault scheme without additional precautions is susceptible against correlation attack. In that regard, we have implemented correlation attacks and empirically substantiated their plausibility.

4.1 Fuzzy Vault Scheme

The fuzzy vault scheme is governed by two basic operations namely *locking* and *unlocking*. The locking and unlocking of the vault are done as follows: Assume that Alice wants to secure her cryptographic key S (a random bit stream) using an arbitrary set of elements A. She selects a polynomial $P(x)$ of degree D and encodes S into the polynomial's coefficients. Encoding can be achieved by slicing S into non-

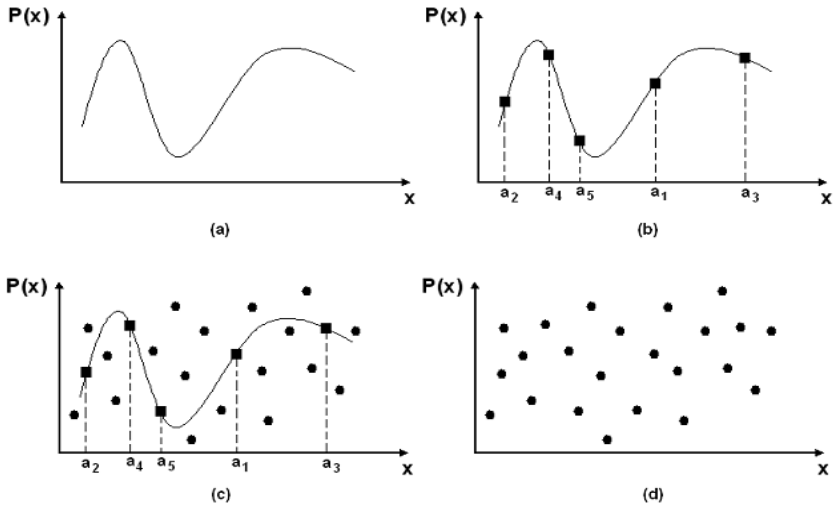

Figure 4.1: Vault *Locking* phase: (a) Create a polynomial by encoding the *Secret* as its coefficients. (b) Project genuine features onto the polynomial: a_i represents the subject's i'th feature. (c) Randomly create chaff points (represented by small black circles) and add to the Vault. (d) Final appearance of the Vault, as stored to the system database.

overlapping bit chunks and then mapping these onto the coefficients. The mapping must be invertible meaning that the coefficients can be unambiguously mapped back to the corresponding bit chunks, which when concatenated, will reconstruct the S. Then, Alice evaluates the polynomial at each element of her set A and stores these evaluation pairs into the set G, where $G = \{(a_1, P(a_1)), (a_2, P(a_2)), ..., (a_N, P(a_N))\}$, $a_i \in A$ and $|A| = N$. Finally, she generates a random set R of pairs such that none of the pairs in that set lie on the polynomial; and she merges the sets G and R into a final set, to obtain the vault, which she then makes public. Note that within the vault, it is not known which points belong to G and which points belong to R. All the steps required to lock a secret in the Fuzzy Vault are graphically represented in Figure 4.1.

Now suppose that Bob has his own set of elements B and he wants to find out ("unlock") Alice's secret locked in the vault. He will be able to do so only if his set B largely overlaps with Alice's A, so as to identify a substantial number of the pairs that lie on the polynomial, from the vault. Given at least $D+1$ pairs that lie on the

polynomial, he applies one of the known polynomial reconstruction techniques (eg. Lagrange interpolating polynomial) to reconstruct the polynomial and thus extracts her secret S. Notice that if Bob does not know which of the points of the vault lie on the polynomial, it should be computationally infeasible for him to unlock the vault.

The fuzzy vault scheme enables privacy protecting matching. Assume the following scenario: Alice locks her telephone number using her favorite music list. She makes her vault public with the hopes to find someone else who shares similar music preference. If Bob has substantially similar music list he will be able to unlock the vault and give a call to Alice. In the above scenario, Alice is protected for unintended calls (not disturbing her privacy) [13].

Whereas perturbation of a single bit in a key of a classical cryptosystem (eg. AES, RSA [72, 73]) hinders decryption completely, the fuzzy vault allows for some minor differences between the encryption and decryption keys, here the unordered sets used to lock and unlock the vault. This fuzziness is necessary for use with biometrics, since different measurements of the same biometric often result in quite different signals, due to a noise in the measurement or non-linear distortions. Furthermore, for most of the known biometric signals, it is hard to establish a consistent ordering within the measured features. For instance two impressions of the same fingerprint can have substantial distortion (displaced minutiae points) and the number of features may vary between the two impressions (eg. missing/spurious minutiae). On the other hand, it is not straightforward how to implement the fuzzy vault using biometric data, due to the difficulty of matching the template and query biometric signals (i.e. locking and unlocking sets, respectively) especially within the presence of random data (the chaff points).

4.1.1 Fuzzy Vault with Fingerprints

Uludag et al. [29] demonstrated a preliminary implementation of the fuzzy vault scheme using fingerprints. Yang and Verbauwhede [28] also implemented the fuzzy vault with fingerprints, but they made the assumption that rotation and translation

invariant features can be reliably extracted from minutiae, which is difficult in practice. Furthermore, they store reference minutia point along with the vault, which may also leak some information. We will review the system by Uludag et al. as it relates the most to our proposed scheme.

Minutia points of template and query fingerprints were used as locking and unlocking sets, respectively, to lock a 128-bit long data (S) which forms the cryptographic key. More precisely, the values obtained by concatenation of the corresponding x and y coordinates of minutiae points were used as set elements. To make sure that the desired S was unlocked from the vault through an error-prone process, cyclic redundancy check bits (16 bits) were concatenated to S. Then, S, together with its check bits, was divided into non-overlapping chunks (16 bits each), giving the coefficients, of an 8th degree polynomial. To lock the secret, template minutiae set was projected onto this polynomial and random chaff points not lying on the polynomial are added, to form the vault. Based on their empirical estimations, they used only 18 minutia points and 200 chaff points.

To unlock the secret, i.e. reconstruct S, they first match the query minutia set with the abscissa part of the vault and identify candidate points lying on the polynomial. Since $D + 1$ points are required to reconstruct a polynomial of degree D, all possible 9 point combinations of the candidate set are tried for reconstruction, to find the one resulting with the correct check bits. S is successfully unlocked when the check bits verify. Authors report a 79% of correct reconstruction rate with 0% false accept rate.

To bypass the problem of matching the minutiae points and finding an upper bound for the performance of the scheme, the authors have used a fingerprint database where minutia points and the correspondence between template and query fingerprints were established by an expert. During their experiments, the minutiae sets of mating fingerprints were pre-aligned (i.e. rotated and translated) according to the established correspondence, and used as such.

4.2 Fuzzy Vault with Online Signatures

We demonstrate an implementation of the fuzzy vault scheme using online signatures. Signature is a behavioral biometric: it is not based on the physical properties of the individual, such as fingerprint or face, but behavioral ones. Online (dynamic) signatures are captured by pressure-sensitive tablets that extract dynamic properties of a signature in addition to its shape. Dynamic features include the number and order of the strokes, the overall speed of the signature, the pen pressure at each point, etc. and make the signature more unique and more difficult to forge, compared to offline signatures consisting of only the image of the signature.

During either of the vault's phases, the user first provides his/her signature using a pressure sensitive tablet. We use the event points of a signature (here after called event points or minutiae points, to make a parallel to fingerprint minutiae) and use these points as locking or unlocking sets. Event points of a signature consists of crossings, endings and places of high curvature. Each such event point is a two dimensional point (i.e. contains x and y coordinates) defined in the Cartesian space of the pressure sensitive tablet. Figure 4.2 demonstrates an example, where event/minutiae points are extracted for the provided signature. The decision to use this representation was mainly due to easily extending the available fuzzy vault implementation with fingerprints to handle online signatures. However, Brault and Plamondon [74] report that curved segments of a signature are making it more complex and difficult to forge, thus our signature minutiae points do possess some of the most important information about a signature.

For the time being minutiae points were marked by experts, i.e. they were not extracted automatically. This was done in order to measure true performance of the vault, i.e. reduce error propagation which could be introduced by an imperfect minutiae extraction algorithm.

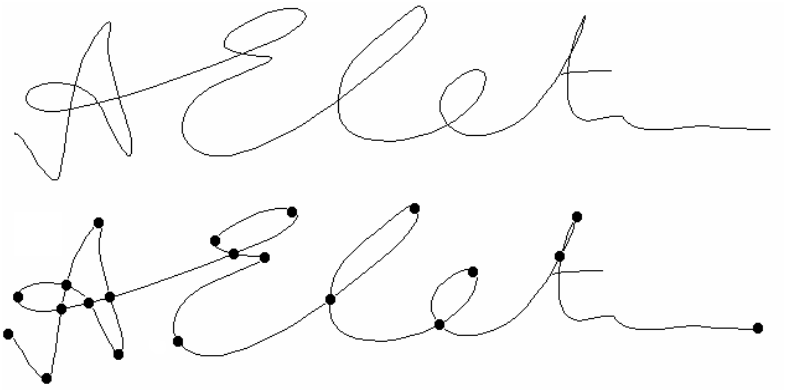

Figure 4.2: A genuine signature (top) and minutiae points marked for that signature (bottom).

4.2.1 Vault Locking

We adopt the notation of Juels et. al [13], while describing the locking and unlocking algorithms. First, a polynomial P of degree D is created by encoding a secret key S into the polynomial's coefficients ($P \leftarrow^{D} S$ denotes the encoding). The polynomial degree was fixed at 9 and didn't change during our experiments. Minutiae point coordinates are then concatenated ($a = x_i|y_i$ denotes the concatenation) and projected onto the polynomial, to give $P(a)$. Minutiae coordinates and their corresponding projections constitute the locking set of the vault. In the pseudo code of the locking algorithm, N denotes the cardinality of the minutiae point set.

Then, chaff points are created as described in the Section 4.1. We randomly select chaff points such that they don't intersect with any other vault point and that they don't lie on the polynomial P. In the pseudo code implementation of the locking algorithm (depicted in the Figure 4.3), K denotes the overall number of the points in the vault such that $D < N << K$; and V is the constructed vault.

Although not explicitly denoted in the locking algorithm, special attention must be paid to situations where chaff points are generated too close to genuine points or other chaff points. In the first case, it may reduce vaults unlocking performance

Public parameters: A field F, a degree D of the polynomial.
Input parameters: Minutiae point set M $\{(x_i, y_i)\}_{i=1}^{N}$ and a secret key S.
Output: A set V of points $\{(x_i, y_i)\}_{i=1}^{K}$, a cryptographic hash of S, $hash(S)$.

Locking Algorithm

$V = \emptyset$;
$P \leftarrow^{D} S$;
for i= 1 to N do
    $a = x_i|y_i$; where $(x_i, y_i)\epsilon M$.
    (x, y)=$(a, P(a))$;
    $V = V \bigcup (x, y)$;
for i= N+1 to K do
    randomly generate (x, y) s.t. $(x, y) \notin V$ and $y \neq P(x)$
    $V = V \bigcup (x, y)$;
Output V and $hash(S)$;

Figure 4.3: A fuzzy vault locking algorithm using signature minutiae point set.

since a chaff point located in close proximity to a genuine point may be mistakenly matched during the unlocking phase. On the other hand, closely generated chaff points may leak information if a malicious attacker knows the closest possible distance between two genuine points. Additionally, chaff points must be homogeneously distributed in the vault space. Otherwise, ill-generated chaff points may leak information, enabling a malicious attacker to reduce his search space and decrease the vault's strength. Figure 4.4 shows a sample Vault which is generated within this system.

4.2.2 Vault Un-Locking

During the unlocking phase, the correspondence between the points of the unlocking minutiae set and those of the vault must be determined. Although there are numerous point matching algorithms, we used exhaustive matching to reduce the error that may be introduced by the matching algorithm. Exhaustive matching is performed by applying all possible rotations and translations (in the Vault space) to the unlocking set, in order to find the alignment with the greatest number of

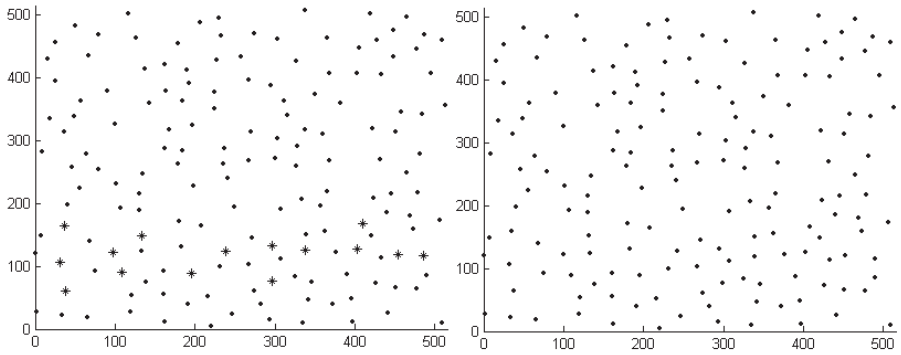

Figure 4.4: The locking of the Fuzzy Vault using on-line signatures: genuine points (stars) and chaff points (dots) are represented differently (left) for the sake of clarity. The actual vault as it is stored to the system's database (right).

matching points. In the scope of the unlocking algorithm illustrated in the Figure 4.5, the function $transform$ takes a point set M and transforms it according to the rotation (r) and translation (t) parameters, while the $\bigcap$ indicates a match with some tolerance. Figure 4.6 shows the result of matching a genuine (left) and a forgery (right) minutiae set with the vault (matching points are circled). As can be seen, while the genuine unlocking set substantially overlaps with the vault's locking set (i.e. genuine points), the forgery set did not.

As a result of matching, we obtain a candidate set of points (C) which are then used for decoding the secret. The decoding is performed using the Lagrange polynomial interpolation method and denoted by $reconstruct(f_i, D)$. During each iteration of the decoding phase, we select $D+1$ points (where D is the degree of the polynomial and f_i indicates selected points) from the candidate set C ($comb(C, D+1)$ indicates a set of all available $D+1$ point combinations) and use them to decode the secret, as described in the Section 4.1. We calculate the hash of the decoded secret and compare it with the one stored in the system's database. Decoding phase terminates when both hash values match (the secret is revealed) or when a maximum number of unlocking attempts (T) is reached (the unlocking failed).

Public parameters: A field F, a degree D of the polynomial, a set T of translations, a set R of rotations.
Input parameters: Query minutiae point set M $\{(x_i, y_i)\}_{i=1}^{N}$, a vault V, a $hash(S)$.
Output: The secret S or $\emptyset$.

Unlocking Algorithm

$C = \emptyset$;
for all r ϵ R do
  for all t ϵ T do
    $M' = transform(M|r, t)$;
    $C' = M' \bigcap V$;
    if $|C'| > |C|$
      $C = C'$;
S'=Decode(C,hash(S));
Output S';

Public parameters: Maximum reconstruction trials T, polynomial degree D.
Input parameters: A polynomial evaluation set C $\{(a_i, P(a_i))\}_{i=1}^{N}$, a $hash(S)$.
Output: A secret S or $\emptyset$.
Decode Algorithm

for i=0 to T do
  f_i ϵ comb(C,D+1) ;
  S'=reconstruct(f_i,D);
  if $hash(S') == hash(S)$
    Output S';
  else
    $S' = \emptyset$;
Output S'

Figure 4.5: A fuzzy vault unlocking algorithm using signature minutiae point set.

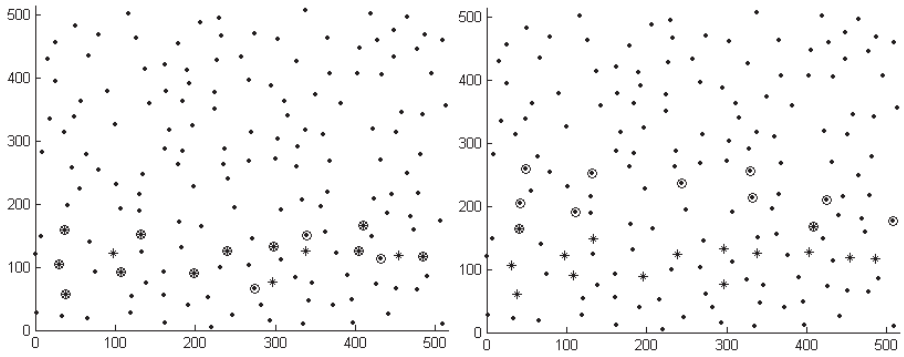

Figure 4.6: Fuzzy Vault Matching using on-line signatures: genuine (left) and forgery (right) minutiae sets are matched with the Vault, respectively. Matched Vault points are circled. For the sake of clarity, minutiae (stars) and chaff (dots) points are represented differently.

4.2.3 Experiments

The system performance was evaluated using sample signatures supplied by 10 subjects enrolled to our system. All signatures in our dataset were acquired using the same tablet and sampled at 100 sample points per second rate. Each subject supplied 4 genuine signatures for a total of 40 signatures. There were no constraints on how to sign, nor was any information given about the working principles of the system, so that the subjects signed in their most natural way.

All possible combinations of 2 signatures out of 4 reference signatures supplied by each subject, were used to measure the vault's genuine performance (i.e. correct unlocking rate of the vault). Thus, 6 such signature pairs were obtained for each user and totally 60 of such pairs were obtained for the whole dataset.

To collect skilled forgeries, we added a signing simulation module to our system. The simulation module animates the signing process of a given signature so that the forger could see not only the signature trajectory, but also the signing dynamics (speed and acceleration). Forgers had a chance of watching the signature's animation several times and practice tracing over the signature image a few times before forging it. We have collected 30 skilled forgeries (3 forgeries for each subject), following the above mentioned protocol. Each forgery signature was paired with each of the

corresponding reference signatures (4 such pairs per forgery signature) and used as such during the locking and unlocking phases, where the reference signature was used to lock and forgery signature to unlock the vault, respectively. A dataset of 120 such pairs was created and used to measure the weakness of the vault against attacks where the forger practiced the locking signature.

We obtained 8.33% failure rate to unlock the vault when a genuine signature was presented (which can be considered as false reject rate) and 2.50% false unlocking rate when the vault was attempted to be opened using forgery signatures (which can be considered as false accept rate). Obtained results are promising: most of the failures to unlock the vault with genuine signatures are due to the high variability within the reference signatures supplied by the user. On the other hand, the false unlocking rate obtained used forgery signatures is due to the fact that only minutiae points, and not the dynamics of the signature, were used. The false accept rate can be reduced by increasing the polynomial degree, of course at the expense of some increase in FRR. Conversely, FRR can be lowered by more efficient chaff point generation, which could assure that number of genuine points matching chaff points doesn't exceed a certain threshold. On average, it took approximately 30 seconds to unlock a vault with it's corresponding genuine signature. Throughout the test, a notebook computer with Intel Celeron (M) 1.5GHz and 512 megabyte of RAM hardware configuration was used. All algorithms were implemented using Matlab.

4.3 Secret Sharing Using Biometric Traits

In biometric based authentication, biometric traits of a person are matched against his/her stored biometric profile and access is granted if there is sufficient match. However, there are other access scenarios that require the participation of multiple previously registered users for a successful authentication or to get an access right. There are cryptographic constructs generally known as **secret sharing** schemes, where a secret is split into shares and distributed amongst the participants in such a way that it is reconstructed/revealed only when the necessary number of share

holders come together. The revealed secret can then be used for encryption or authentication (if the revealed key is verified against the previously registered value).

We propose a method for the biometric based secret sharing. Instead of splitting a secret amongst participants, as is done in a cryptographic secret sharing, a single biometric construct is created using the biometric traits of the participants. During authentication, a valid cryptographic key is released out of the construct when the required number of genuine participants present their biometric traits. The method uses the *fuzzy vault* construct suggested by Juels et al. [13].

4.3.1 Cryptographic Secret Sharing

Shamir first proposed a method (known as *threshold scheme*) for secret sharing, which can be defined as follows in the most general form [75]:

Definition: Let T and N be positive integers with $T \leq N$. A (T, N)**-threshold scheme** is a method of sharing a secret S among a set of N participants such that any subset of T participants can reconstruct the secret S, but fewer than T participants cannot.

Shamir's scheme is based on encoding a secret S as the constant term of a polynomial of degree $T-1$, whose coefficients, which are kept secret, are smaller than a certain prime p. Hence, the polynomial can be given as follows:

$$P(x) = S + c_1 x + c_2 x^2 + \ldots + c_{T-1} x^{T-1} \pmod{p}$$

The polynomial is then evaluated for N randomly chosen distinct integers $x_1, x_2, \ldots x_N \pmod{p}$ and each participant is given a pair (x_i, y_i) with $y_i \equiv P(x_i) \pmod{p}$, where $i = 1, 2, \ldots, N$. The prime p is known to all participants, but the polynomial $P(x)$ is kept secret.

The polynomial, $P(x)$, whose degree is $T-1$ can be reconstructed if any T distinct evaluation pairs $(x_{L_1}, y_{L_1}), \ldots, (x_{L_T}, y_{L_T})$ are known, using *Lagrange interpolation method*. Consequently, any T participants from the set of N participants can come together and reconstruct $P(x)$ and hence find out the secret S, which is

the constant term. The secret can be used in many ways in a security context such as encryption, decryption, authentication, signatures, etc.

In this proposed scheme, the distinct integers x_i $(i = 0, \ldots, N)$ where the polynomial $P(x)$ is evaluated, are no longer chosen at random but derived from the biometrics of the participants; the evaluated pairs are then kept in a fuzzy vault, as explained in the following sections.

4.3.2 Secret Sharing Using Fuzzy Vault

In this section we describe how to implement the secret sharing scheme [75], using the fuzzy vault scheme suggested by Juels et al. [13]. As the most general form, our secret sharing scheme implements a *threshold scheme*, revealing a secret when a predetermined number of the sharing parties collaborate. Even though extending the fuzzy vault to implement secret sharing is relatively straightforward, we addressed a few non-trivial issues for this extension. Furthermore, this is the first biometrics-based secret sharing scheme, to our knowledge.

Suppose N people who want to share a secret S such that at least T out of these N people are required to reveal the secret, where $T \leq N$. We use their biometric signals G_1 through G_N and the secret S to construct the fuzzy vault as follows: Assume for simplicity that the biometric signal is one dimensional such that $G_i = \{g_{i1}, g_{i2}, ...g_{iL_i}\}$, where i denotes the participant's ID; g_{ij} denotes a scalar measurement in the signal; and L_i denotes length of the signal. We assume that the length of each party's biometric signal is greater than or equal to some fixed length K, i.e. $L_i \geq K$. We select K out of each of the L_i measured features and the rest are discarded.

Next, we pick a polynomial $P(x)$ of degree D such that $(T-1)K \leq D \leq TK-1$, where $P(x) = c_D x^D + c_{D-1}x^{D-1} + \cdots + c_1 x + c_0$, c_i denotes the i'th coefficient of the polynomial. Note that to reconstruct such a polynomial, biometric features of at least T parties are required, as intended. The coefficients may be determined as described in Section 4.3.1, where S can be used as the constant term. The other way is to divide S into non-overlapping bit chunks and then map these chunks onto

the coefficients.

After fixing the polynomial, we compute projections of each participant's biometric signal (i.e. G_i's) onto the polynomial, and obtain evaluation pair sets denoted by $A_i = \{(g_{ij}, P(g_{ij}), \ldots (g_{iK}, P(g_{iK})\}$. Next we create a set C of M random chaff points, where $C = \{(r_1, P(d_1)), \ldots, (r_M, P(d_M))\}$, where r_i and d_i are random numbers. Random points are generated in such a way that non of them coincide with genuine points or lie on the polynomial; i.e. $r_l \neq d_l$ and $r_l \neq g_{ij}$. Finally, the genuine sets A_i's and the set of chaff points C are merged and shuffled to constitute the fuzzy vault, V. The number of chaff points M must be selected such that identifying the required number of genuine evaluation pairs (i.e. $D+1$ of them) without possessing the required number of genuine biometric traits will be computationally hard for an adversary or a malicious group of genuine participants (a group smaller than T).

When a group of T genuine participants decide to reconstruct the secret, they submit their corresponding biometrics that are used to match the abscissa part of the fuzzy vault's points, one person at a time. Then, the identified points of the vault are used for the polynomial reconstruction and the secret will be extracted, in the way described in Section 4.1.

Note that the template minutiae of a person can match at most K genuine points (which is enough to reconstruct the selected polynomial) and possibly some chaff points and some of the other parties' genuine points. The matched chaff points are handled the same way as in the fuzzy vault implementation, by examining some subset of the matched points and checking the hash code. Matching other genuine points is more important, because it would leave fewer than K points to be matched by the true owner, as well as possibly compromising the sharing scheme by causing a situation where less than T genuine participants could match $D+1$ genuine vault points. This is handled by spacing apart each set of genuine K features from each other. For instance, if secret sharing is performed using fingerprint minutiae, as in this work, then each K minutiae can be translated to different parts of the vault space. Alternatively, the degree of the polynomial can be selected closer to it's possible upper bound (i.e. $TK-1$) to lower the risk of revealing the secret to less

than T genuine participants. However this can also possibly increase the false reject rate of the sharing scheme.

This problem can also be addressed during the selection of K of the features. which can be done in a couple of different ways: according to the significance of the corresponding features (i.e. select K most discriminative features for each party); or randomly, if each feature conveys an equal amount of information. During the selection process it may happen that some of the K features selected for different parties do clash (i.e. $g_{im} = g_{jn}$), which would be undesirable because it would result in fewer than expected number of genuine points within the fuzzy vault, as mentioned above. To resolve this, different strategies could be adopted as suitable: i) if available, replace g_{in} or g_{jm} with another feature of the corresponding party, ii) transform (eg. affine) g_i or g_j so that they don't overlap, iii) reduce K, or iv) reduce degree of the polynomial. Selection of a strategy will depend on a biometric signal used, a number of participants (N) and the security level. In particular, in our implementation with fingerprints, we selected the points around the center of mass of the corresponding minutiae set (to reduce the areas which are more prone to occlusion) and translated them in the vault space to avoid clashes (see Sec. 4.3.3 for details).

4.3.3 Implementation

In this section we present the implementation of the secret sharing scheme using fingerprints. First, we describe our own implementation of fuzzy vault for fingerprints and then demonstrate its utilization for secret sharing. Our implementation of the fuzzy vault follows similar locking and unlocking steps as used in Uludag et al. However our system is fully automatic: corresponding locking and unlocking sets are not pre-aligned, which makes it suitable for real life applications.

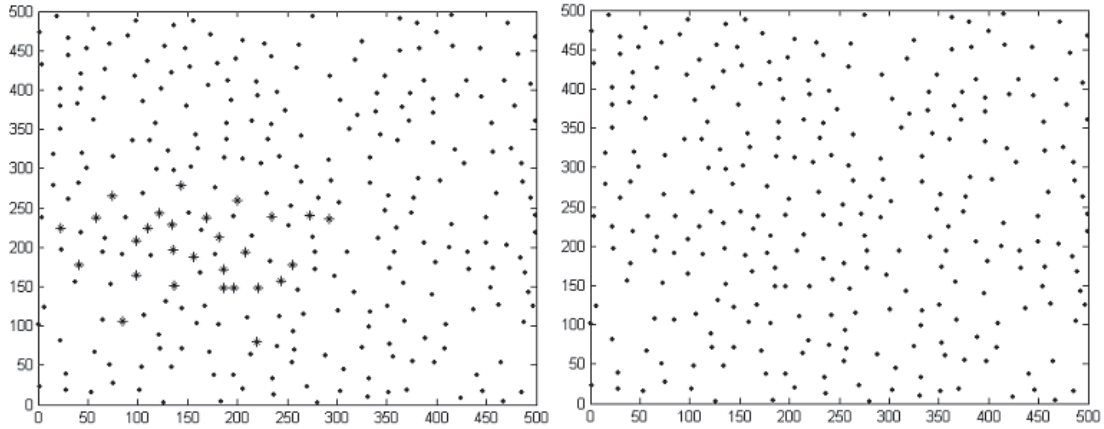

Figure 4.7: The locking of the Fuzzy Vault using fingerprints: minutiae (stars) and chaff (dots) points are represented differently (left) for the sake of clarity. The actual vault (right) as it is stored to the system database.

Fuzzy Vault Implementation

Our implementation of the fuzzy vault using fingerprints is based on that of Uludag et al. [29] presented in 4.1; however, ours has an automatic matcher and other small modifications, as explained in this section.

In addition to the positional information, each minutiae is also associated with a number of other features, such as the angle of the ridge at the minutiae location, the gray level profile around minutia, as well as different counts of and distances to neighboring minutiae, etc. However, features besides the x,y-coordinates may leak information (i.e. reveal genuine points in the vault), and thus are not used. We concatenate the x and y coordinates of minutiae points and project these onto the secret polynomial, as described in Section 4.1. Minutiae coordinates and their corresponding projections constitute the locking set of the vault.

Then, random chaff points are created, again as described in 4.1. It is worth to mention that chaff points must not be too close to genuine points and each other; there must be at least a distance equal to the inter-ridge distance (approximately 16 pixels in 500 dpi images). Also chaff points must be homogeneously distributed in the vault space. Otherwise, ill-generated chaff points may leak information, enabling

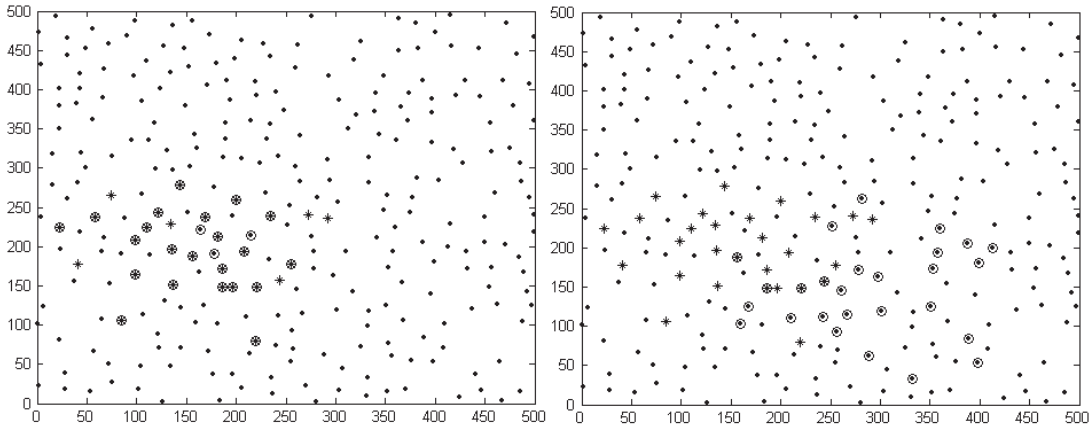

Figure 4.8: The matching of the Fuzzy Vault with genuine (left) and forgery (right) query minutiae sets. Matched vault points are circled.

a malicious attacker to reduce his search space and decrease the vault's strength. Figure 4.7 shows a sample vault which is generated within this system.

During the unlocking phase, the correspondence between the minutiae points of the unlocking set and those of the vault must be determined. Although there are numerous point matching algorithms, we used exhaustive matching to reduce the error that may be introduced by the matching algorithm. Exhaustive matching is performed by applying all possible rotations and translations (in the vault space) to the unlocking set, to find the alignment with the most number of matching points. Figure 4.8 shows the result of matching genuine (left) and forgery (right) minutiae sets with the vault (matched vault points are circled). As can be seen, while genuine unlocking set substantially overlaps with the vault's genuine points, the forgery set did not.

As a result of matching, we obtain a candidate set of points consisting of matched genuine and chaff points. During each iteration of the decoding phase, we select $D + 1$ (where D is the degree of the polynomial) points from the candidate set and use them to try to decode the secret as described in Section 4.1. The polynomial degree was fixed at 9 and didn't change during testing. During the locking phase, a cryptographic hash of the secret is also stored in the system database, along with the

vault points. In each decoding attempt, we calculate the hash of the decoded secret and compare it with the one stored in the system's database. The decoding phase terminates when both hash values match (secret is decoded) or when maximum number of iterations is performed (secret not revealed).

Secret Sharing Implementation

In this section we demonstrate the utilization of the fuzzy vault described above, for secret sharing. In our sample scenario, 3 users share a secret such that at least 2 of them must present their fingerprints to reveal the secret.

During the locking phase, for each participant we select 13 of his/her minutiae points, discarding the rest. The selection of minutiae is performed around the center of mass of the corresponding fingerprint, to reduce possible matching errors caused by occlusions. During the selection process, we follow the guidelines introduced in Section 4.3.2. Hence, a total of 39 (13x3) minutiae are used in the locking set.

As described in Section 4.3.2, the degree (D) of the polynomial to which the secret is to be encoded must satisfy the following condition $(T-1)K \leq D \leq TK-1$, where T denotes the minimum number of participants required to reveal the secret and K denotes the number of features each participant possesses. Thus, any degree between 13 and 25 will satisfy the requirements; we used a degree of 17. The locking set is then projected onto the polynomial, forming the vault's genuine points.

In the next step, random chaff points are generated, as was done in Section 4.3.3. We should mention that, during chaff point generation, discarded genuine minutiae are also considered as if they were present in the vault. This is done to reduce false reject rate, since chaff points generated close enough (less than inter-ridge distance) to places where the discarded minutiae were located may match with minutiae of the unlocking set.

The scheme requires at least participants to present their minutiae. During the unlocking phase, the minutiae set of each of the two participants is matched sequentially with the vault, as described in Section 4.3.3. Matched vault points are discarded before subsequent match. Figure 4.9 demonstrates the vault (left) and

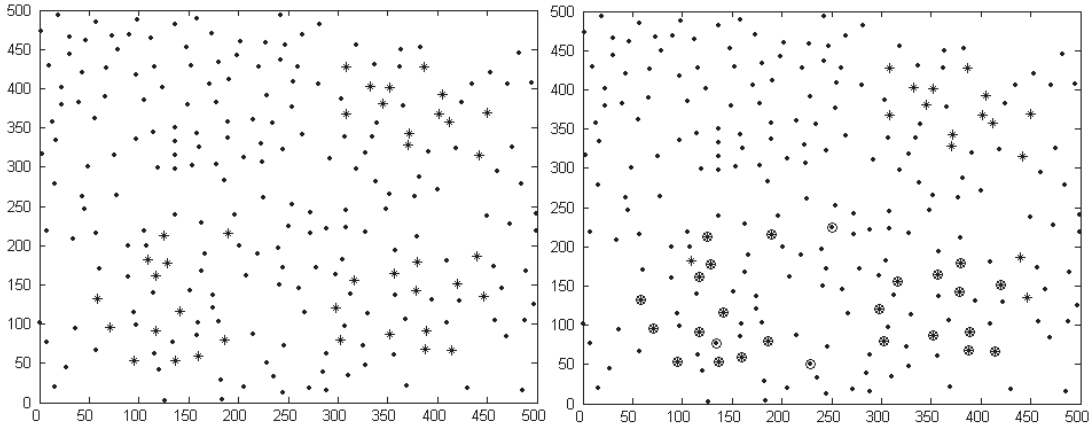

Figure 4.9: Secret sharing using fuzzy vault. The vault is created using fingerprint minutiae of 3 different users (left). The vault is matched using query minutiae of two genuine users (right).

the result of matching the vault with unlocking minutiae sets (right). As a result of matching, a candidate set of points is obtained, which is then used for decoding the secret as is done in Section 4.3.3.

4.3.4 Experiments

We used a dataset of 200 different fingerprints with 2 imprints per finger (400 imprints total), to test both the success of the vault implementation and the success of the secret sharing scheme. Minutiae locations were manually labeled by experts.

To test the fuzzy vault system's performance, we tried to unlock a vault with either the minutiae set of the matching fingerprint (genuine) or the minutiae set of an non-matching fingerprint (forgery). Minutiae correspondences were found automatically by the system during the unlocking process. In these tests, we achieved a 4% false accept rate (FAR) and a 6.5% false reject rate (FRR). In comparison, the results of Uludag et al. were 21% FRR and 0% FAR using *manually aligned* minutiae sets. The false accept rate can be reduced by increasing the polynomial degree, of course at the expense of slight increase in FRR. Conversely, FRR can be lowered by more efficient chaff point generation, which could assure that number of genuine

points matching chaff points doesn't exceed a certain threshold. On average, it took approximately 40 seconds to unlock a vault with it's corresponding genuine fingerprint. Throughout the test a notebook computer with Intel Celeron (M) 1.5GHz and 512 megabyte of RAM hardware configuration was used. All algorithms were implemented using Matlab.

The secret sharing scheme's performance was tested using the same dataset. To share a secret, fingerprints of 3 participants are required to lock the vault. We randomly created 400 triplets of fingerprint imprints to lock the vaults. To reveal the secret, any 2 out of 3 participants must submit their fingerprints. For each triplet, there are 3 different combinations of 2 genuine participants. Thus, a total of 1200 unlocking attempts were made with genuine minutiae sets, to measure FRR. To test the performance of the system against forgeries (FAR), we simulated the case where only one person tries to unlock the secret, by using one genuine fingerprint and one random forgery fingerprint. 1200 unique unlocking attempts were performed to accomplish this task. In these tests, we achieved 0% of FAR (only one of the genuine participants attempts to unlock the vault) and 10.5% FRR (2 of the genuine participants attempt to unlock the vault).

4.4 Realization of Correlation Attack Against the Fuzzy Vault Scheme

It was recently claimed that the fuzzy vault scheme is susceptible to correlation based attacks, if two or more fuzzy vaults created using the same biometric data (e.g. two impressions of the same fingerprint) were available. It suggests that correlating them would reveal the biometric data hidden inside [76].

In this part of the thesis, we performed correlation attacks against a database of 400 fuzzy vaults (200 matching pairs), created using our implementation described in Section 4.3.3. Given two matching vaults, we could successfully unlock 59% of them within a short time. Furthermore, it was possible to link an unknown vault to

a short list containing its matching pair, for 41% of all vaults. These results prove the claim that the fuzzy vault scheme without additional security measures is indeed vulnerable to correlation attacks.

4.4.1 Attacks on Fuzzy Vault

Scheirer et al. [76] suggests a number of attacks targeted on the fuzzy vault scheme. They classify suggested attacks into 3 groups: i) attacks via record multiplicity, ii) stolen key-inversion attack, and iii) blended substitution attack.

Attacks via record multiplicity assume that an attacker intercepts multiple enrollments/encodings which are created using the same biometric data (e.g. two fuzzy vaults created using imprints of the same fingerprint, but different chaff points). Scheirer et al. claimed that correlation of two such enrollments may reveal the biometric data encoded within each enrollment, suggesting that a correlation peak must occur at the point where biometric data of corresponding encodings overlap the most.

Stolen key-inversion attack assumes that an attacker obtains a secret key which is released upon the presentation of a genuine biometric to a corresponding system. The attacker could obtain that key by means of social engineering or weak coupling between different modules of the system, etc. Then, the attacker may retrieve the biometric data from its corresponding encoding, by utilizing this key. In the case of fuzzy vault, it is a straightforward task to identify genuine and chaff points given the key (i.e. the polynomial coefficients) is known: Using the key, one would compute $p(x)$ for each given x and verify which of the vault points (chaff or biometric) actually have the correct $p(x)$, hence correspond to the biometric.

Finally, the blended substitution attack considers the scenario where a malicious attacker injects his own data into someone's template. After such injection, both genuine and malicious users will be positively authenticated against the same enrollment record. In the case of the fuzzy vault scheme, during the vault construction the attacker would insert his own minutiae points, possibly using his own secret (i.e. polynomial). When the attacker later presents his fingerprint for the verification,

the genuine user's minutiae points will act as chaff points and the attacker will get authenticated. While the implementation of this attack may not be straightforward, it is obvious that the fuzzy vault may be susceptible for this type of attack. In fact, Kholmatov et al. [77] exploited this additive property of the fuzzy vault scheme to implement a biometric based secret sharing.

Scheirer et al. only suggested the aforementioned attacks, without providing any particular implementations. As the success of stolen key-inversion attacks against the fuzzy vault scheme is mostly obvious, the success of the other two attacks must be substantiated. In this work, we address this issue and empirically assess the vulnerability of the fuzzy vault scheme against record multiplicity attacks.

Note that recently Nandakumar et al [78] described and implemented a fuzzy vault that is hardened using passwords, to eliminate some of the weaknesses of the fuzzy vault scheme. Specifically, they construct the fuzzy vault after transforming (translation and rotation) the fingerprint minutiae, effectively eliminating correlation-based attacks. However, their scheme causes some increase in the False Reject Rate due to the matching being carried in the transformed space. Furthermore, considering the case where the password may be compromised, the vulnerability of the vault remains to be quantified.

4.4.2 Implementation of Correlation Based Attacks

We suggest to assess the feasibility of the correlation attack under three different scenarios: i) given two fuzzy vaults locked using the same biometric trait but different chaff points and different secrets (i.e. different polynomials), is it practically possible to reconstruct the secrets encoded in their corresponding vaults? ii) given a fuzzy vault belonging to an unknown person and a set of vaults linked to their corresponding identities, what is the success rate of correctly identifying the unknown person using the correlation attack? iii) given two sets of vaults, how likely is it to find the correspondences of the vaults in these sets? Note that the second scenario is a subtask of the third.

We carried our experiments on the fuzzy vaults created using our fingerprint-

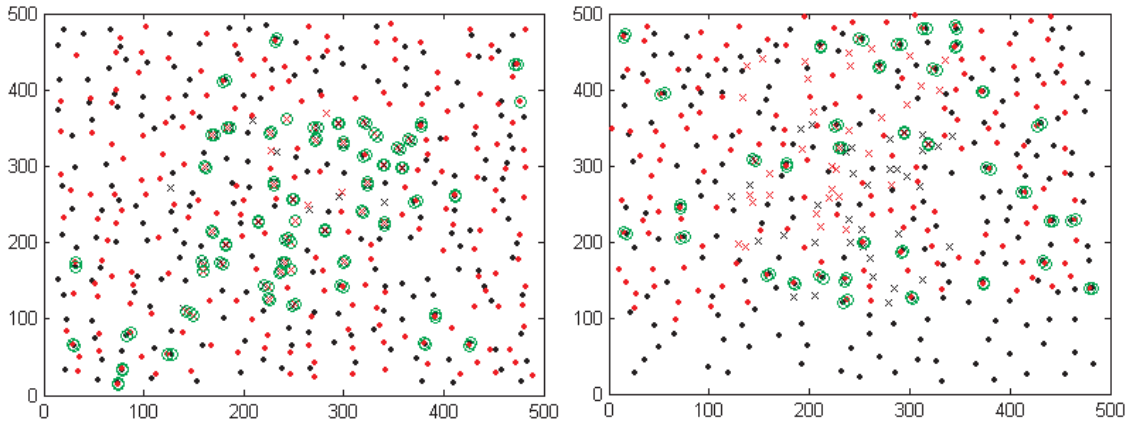

Figure 4.10: Alignments of two vaults, created using different impressions of the same fingerprint (left) and completely different fingerprints (right). Crosses represent fingerprint minutiae, dots identify chaff points. Minutiae and chaff points of a corresponding vault are colored by the same color (red or black) and matching points are also circled.

based fuzzy vault construction scheme [77], described in Section 4.3.3. The fingerprint database used to construct the fuzzy vaults consists of 400 fingerprint images (2 different impressions for each of 200 different fingers) acquired with an optical sensor and with manually labeled minutiae points. Using this database, we created a fuzzy vault from each fingerprint (a total of 200 fuzzy vaults pairs). Each fuzzy vault is locked using different chaff points and different secrets (i.e. different polynomials). The number of chaff points added to each vault was 200 and the polynomial degree was set to 8, as used in [29, 77].

4.4.3 Unlocking Two Matching Fuzzy Vaults

In order to assess the feasibility of the first scenario, we matched each vault pair using exhaustive matching, seeking the best alignment between two vaults over all of their relative rotations and translations. The best alignment is decided as the one which maximizes the number of matching points between the two vaults. We follow the notation established in the Section 4.2 while outlining the pseudo code for the vault cracking algorithm depicted in the Figure 4.11, which indeed parallels the

vault unlocking algorithm described in the mentioned section. Figure 4.10 depicts the alignments of two vaults created using the same (on the left) and different (on the right) fingerprints. As can be seen, genuine points align very well for the vaults created using the same fingerprint, while that is not the case for non-matching vaults.

After the alignment, we obtain two candidate point sets (C_A, C_B), one for each of the two vaults, containing a mixture of genuine and chaff points. The vault(s) can be unlocked if i) the number of matching minutiae points (M) within a candidate set is sufficient to decode the polynomial of degree D (i.e. $M \geq D + 1$) and ii) the total number of matching points (N) is not too large with respect to M, so that the brute force attack (i.e. trying all possible combinations of $D + 1$ out of N points) to reconstruct the polynomial is computationally feasible. The expected number of attempts (A) to unlock one of the vaults can be formulated as follows:

$$A = \frac{\begin{pmatrix} N \\ D+1 \end{pmatrix}}{\begin{pmatrix} M \\ D+1 \end{pmatrix}}$$

During our experiments, we observed that on the average there are 43 matching points (i.e. N) between corresponding vaults, where 22 out of these points are minutiae points (i.e. M). Based on the above formulation, 1133 attempts are required on the average to reconstruct corresponding vaults and the whole process (exhaustive matching and decoding attempts) takes approximately 50 seconds for a non-optimized Matlab implementation on a PC with 3GHz CPU and 8 GByte RAM. Notice that once we unlock one of the vaults, separating chaff and minutiae points is done, and unlocking the second vault is straightforward.

Using this method, we could successfully reconstruct 59% of the corresponding fuzzy vault pairs (i.e. 118 out 200) created using our fingerprint database. This success rate is enough to conclude that the fuzzy vault is indeed vulnerable against

Public parameters: A field F, a degree D of the polynomial, a set T of translations, a set R of rotations.
Input parameters: Two vaults V_A and V_B, cryptographic hashes of their secrets $hash(S_A)$ and $hash(S_B)$.
Output: (S_A, S_B) or $\emptyset$.

Vault Cracking Algorithm

$C_A, C_B = \emptyset$;
for all r ϵ R do
  for all t ϵ T do
    $V'_A = transform(V_A|r, t)$;
    $(C'_A, C'_B) = V'_A \bigcap V_B$;
    if $|C'_A| > |C_A|$
      $C_A = C'_A$;
      $C_B = C'_B$;
S'_A=Decode(C_A,hash(S_A));
S'_B=Decode(C_B,hash(S_B));
Output (S'_A, S'_B);

Public parameters: Maximum reconstruction trials A, polynomial degree D.
Input parameters: A polynomial evaluation set C $\{(a_i, P(a_i))\}_{i=1}^{N}$, a $hash(S)$.
Output: A secret S or $\emptyset$.
Decode Algorithm

for i=0 to T do
  f_i ϵ comb(C,D+1) ;
  S'=reconstruct(f_i,D);
  if $hash(S') == hash(S)$
    Output S'
  else
  $S' = \emptyset$;
Output S'

Figure 4.11: An algorithm for unlocking two matching fuzzy vaults.

correlation attacks. In order to increase the success rate, one could also try to use alignments where the number of matching points exceed certain threshold, instead of using only the best alignment.

4.4.4 Correlating Two Databases

The success rates of last two scenarios mentioned in Section 4.4.2 are assessed by matching each fuzzy vault with the rest of the available vaults (399 matches for our database), to simulate the situation where one vault would be matched to all the vaults in a separate database.

In matching one vault against all others, it is reasonable to expect that the alignment with the corresponding vaults (i.e. those created using different impressions of the same fingerprint) would result in the highest number of matching points (i.e. N), compared to the alignments with non-matching vaults. We tried to validate this expectation and aligned each vault with all of the remaining vaults in the database and picked one with the highest number of matching points for decoding. This process returned the matching vault as the top-choice for 24% of all the vaults (96/400). When we considered top-10 choices, the matching vault was found for 41% of all the vaults. Since unlocking attempts for a vault takes a short time, an attacker who wants to find a matching vault can find the best aligned vaults and try to decode them all in turn, succeeding with a high probability in a reasonable time ($50sec \times 10$ attempts $= 8min$, in the worst case for our implementation).

These experiments support the claim that the fuzzy vault scheme without additional security enhancing measures such as the one suggested by Nandakumar et al [78], is vulnerable to correlation attacks. On the other hand, such attacks will not succeed if the fuzzy vault is locked using behavioral traits such as signature or voice. In such case, the user will have to lock vaults used for different applications with distinct instances of his/her behavioral traits (eg. using different signatures or vocal pass phrases for different applications).

4.5 Summary and Conclusion

Bio-cryptosystems combine cryptography and biometrics to take advantage of the benefits of both fields: while biometrics provide non-repudiation and convenience, traditional cryptography provides adjustable levels of security and can be used not just for authentication, but also for encryption. The fuzzy vault has emerged as a promising method for such synergy. The scheme is based on binding the biometric template with a secret key and scrambling it with a large amount of redundant data, such that it is computationally infeasible to extract the secret key without possession of the biometric trait. Fuzzy vault doesn't require ordered representation of a biometric and it can tolerate variations within a biometric, up to some extent.

In the context of this work, we demonstrated a fully automatic and practical implementation of the fuzzy vault scheme using fingerprints and online signatures. Even though our implementations are relatively straightforward extensions of the implementation by Uludag et al. [29], the issues encountered in implementing the fuzzy vault with online signatures were non-trivial. Besides, it is the first realization of the scheme using online signatures which demonstrated promising performance results.

The secret sharing is a well-known family of cryptographic protocols. It provides guidelines for the release of a secret key upon providing a predefined number of shares required to reconstruct the key. We demonstrated secret sharing using biometric traits by utilizing the fuzzy vault scheme. The resulting scheme enhances traditional secret sharing scheme proposed by Shamir [75], in that it benefits from convenience and non-repudiation provided by biometrics. Even though it is a relatively straightforward extension of the fuzzy vault scheme, the issues encountered in implementing the fuzzy vault with multiple people and implementing the threshold scheme were non-trivial.

Finally, it was recently claimed that the fuzzy vault scheme is susceptible against correlation attacks, however without any proof. In the context of this work, we implemented correlation attacks and empirically assessed the vulnerability of the

fuzzy vault scheme against these attacks.

We implemented the fuzzy vault scheme using fingerprints and found that using a database of 400 fingerprint impressions, we were able to successfully unlock 59% of the vaults created using different impressions of the same fingerprint. Additionally, we showed that for 41% of all the cases, it was possible to link an unknown vault to a small set of vaults containing the matching vault. This results prove the claim that the fuzzy vaults scheme without additional security measures is vulnerable to such attacks.

On the other hand, such attacks will not succeed if the fuzzy vault will be locked using behavioral traits such as signature or voice. In such case, the user will have to lock vaults used for different applications with distinct instances of his/her behavioral traits (eg. using different signatures or vocal pass phrases for different applications).

Chapter 5

Individuality Model for On-line Signatures

In this part of the thesis, we study the discriminative capability of online signatures as it directly relates to the success of using online signatures in the available privacy preserving biometric verification systems such as fuzzy vault and multi-biometric templates approaches.

The discriminative capability of a biometric is based on its individuality/uniqueness and is an important factor in choosing a biometric for a large-scale deployment or a cryptographic application. Individuality studies have been carried out rigorously for only certain biometrics, in particular fingerprint and iris, while work on establishing handwriting and signature individuality has been mainly on feature level. In this part of the thesis, we present an individuality model for online signatures using the Fourier domain representation of the signature. Specifically, using the normalized Fourier coefficients as global features describing the signature, we derive a formula for the probability of coincidentally matching a given signature.

5.1 Introduction

With increased popularity of biometric verification systems, new biometric modalities are introduced every year (e.g. tongue shape). The performance of a biometric verification system is commonly reported using false accept and false reject rates (FAR and FRR, respectively). These performance measures depend on the verification algorithm and more importantly, the particular database used in testing the

system; hence they do not truly represent the discriminative capability of the corresponding biometric data, nor do they provide performance estimates for the cases where corresponding biometric would be deployed to the general public.

The discriminative capability of a biometric modality is based on its individuality/uniqueness across the population and can be measured as the probability of a coincidental match between the biometric data of two different subjects. This probability depends on the concept of *entropy* [79] which denotes the number of bits of information present in a certain message or event. For instance, in a classical password-based system, the security of the system depends on how easy it is to guess a password, which in turn depends on the number of available passwords. As a simple example, assuming a uniform probability model, 4 digit pin-codes used in ATM cards have a 1/10000 chance of being guessed; equivalently, they possess a $-log_2(1/10000) = 14$ bits of *entropy*. Within the context of biometric individuality, the term *guessing entropy* [80, 81] is used in particular, to emphasize that entropy results are obtained considering random guesses. This distinction is more relevant in the context of signatures, or behavioral biometrics in general, where skilled forgeries present a bigger challenge. Nonetheless, it is important to know how easy it would be to guess a random, average online signature without knowing anything about it. This is comparable to guessing someone's ATM pin without knowing anything about the person.

Individuality studies have been carried out empirically or theoretically, for only certain biometrics to the best of our knowledge: some work is done towards establishing handwriting [82, 83], voice [84, 85] and signature [86, 87] individuality, while more comprehensive studies are carried out for fingerprint [88,89] and iris [20,90,91].

Empirical studies consider data from a large and representative population to assess the individuality of the biometric as the minimum dissimilarity between non-matching samples. For instance, Daugman has shown that with respect to a given representation (*IrisCode*) and a matching algorithm, genuine and forgery populations show clearly separated distributions [20], corresponding to a Binomial distribution with 173 bits of freedom; concluding that an iris carries 173 bits of information.

Theoretical studies try to break the dependence upon the current database by considering the probability of match within the whole parameter space of a particular representation. However, they also depend on a particular representation and a matching algorithm (preferably well-accepted ones), and even a database commonly used to estimate certain parameters; hence the distinction between two definitions is not very clear-cut. Nonetheless, there is a distinction which is due to the fact that the estimates calculated in theoretical studies are in general in lower level features (e.g. how far apart can two matching minutiae points are), whereas empirical models estimate the guessing entropy on the higher level features (e.g. how likely it is to match a fixed number of minutiae in a fingerprint). Theoretical studies are very useful in estimating the performance of a verification algorithm when deployed in a large-scale application, because lower level statistics can be obtained more reliably in smaller size databases. In the case of empirical approaches, it is important to obtain a large enough and statistically representative biometric samples. Furthermore, the similarity comparisons may be prohibitively time consuming. On the other hand, theoretical approaches face the difficulty of realistically modeling all available dimensions of a biometric and formulating the likelihood of coincidental match.

It is important to note that any individuality model must be done with respect to a given feature representation, as well as a matching algorithm in order to allow for intra-class variations within the biometric with respect to inter-class variations (variations within the biometric measurements of a person and those between the measurements of a person and those of the forgers for that person). Simply calculating the entropy of a biometric signal without regard to the intra-class variations would result in an unrealistically optimistic entropy measure. Due to this dependence, individuality measurements of biometrics may be refined in time as more powerful features and matching algorithms are found.

While there are some works in the area of handwriting individuality for online and off-line handwriting [82,83] and feature level entropy for online signature verification, the individuality of a signature, or the biometric entropy of a signature as a biometric signal, is not assessed to the best of our knowledge. Once determined, the entropy of

a signature will provide insights on the potential of online signatures as a biometric identity of the individual.

In this study, we present an individuality model for online signatures using the Fourier domain representation of a signature. It should be emphasized that unlike a fingerprint or an iris, a signature may have arbitrary complexity, so this study is concerned with the entropy of an average signature as found in a large database. Also an important distinction needs to be made between entropy, which is the topic of this chapter, and forgeability. As previously brought up by Ballard et al. [92, 93], these two concepts are related but not equal. Entropy in this context refers to how difficult it is to guess someone's signature and compares to random forgery tests, while forgeability refers to how difficult it is to forge someone's signature and is measured by false accept rates.

5.2 Background on Online Signature Verification

Signature verification is split into two areas –*online* and *offline*– depending on the type of available data. Online (dynamic) signatures are captured by special hardware (eg. smart pens or pressure sensitive tablets) which is capable of measuring dynamic properties of a signature in addition to its shape, while the shape is the only available information in offline signatures. Dynamic information (e.g. pen pressure) makes the signature more unique and more difficult to forge. Applications of online signature verification include identity verification in payments using a credit card; authorization of computer users for accessing sensitive data or programs; authentication of individuals for accessing physical devices or buildings; and protection of small personal devices (e.g. PDA, laptop) from unauthorized usage.

Signature verification systems differ both in their feature selection and their decision methodologies. In fact, more than 70 different feature types have been used for signature verification [94–96]. These features can be classified in two types: global and local. Global features are those related to the signature as a whole, including the signature bounding box dimensions, average signing speed, and signing

duration. Local features on the other hand are measured or extracted at each point along the trajectory of the signature. Examples of local features include position, speed, curvature and pressure at each point on the signature trajectory. In [94], some of these features are compared in order to find the more robust ones for signature verification purposes. Other systems have used genetic algorithms to find the most useful features [97].

Genuine signatures of a person often differ in length due to the natural variations in signing speed. The advantage of global features is that there are a fixed number of measurements (features) per signature, regardless of the signature length; this makes the comparison of two signatures a relatively straightforward task. When local features are used, one needs to use methods which are suitable to compare feature vectors of different lengths: for instance the dynamic time warping algorithm [94, 96, 98–100] or Hidden Markov Models [101]. These latter methods are more complicated than relatively simple metrics using global features, but they are usually more successful as well. A comprehensive survey of signature verification can be found in [102, 103].

Due to the differences in databases and forgery qualities, comparing reported performance results has been difficult. The First International Signature Verification Competition (SVC2004), organized in 2004, provided a common test set and tested more than 15 online signature verification systems from industry and academia. The results of this competition indicate state-of-the-art results of 2.6% equal error rate (EER) in skilled forgery detection and 1.85% equal error rate in random forgery detection tasks [104]. While the participants' identities were kept confidential except for the winners, our system, later described in [98], was declared as the winning system for its performance in the skilled forgery test.

5.3 Previous Work on Biometric Individuality

The individuality studies on fingerprint [88, 89] and iris [20] modalities have constituted the starting points of this work: Pankanti et al. [88] provides a broad overview

of research done on fingerprint individuality, as well as proposing their own individuality model. Their model is based on minutiae representation of the fingerprint and individuality is modeled by the probability of false correspondence between two unrelated fingerprints' minutiae: two fingerprints are considered similar if they match in at least k minutiae points after appropriate alignment. In turn, two corresponding minutiae are accepted as matching if both their locations and orientations are substantially similar. The probability of match between two arbitrary fingerprints in at least k minutiae points is modeled by the hypergeometric distribution. Parameters which govern the proposed model are estimated using a large fingerprint database. Using the proposed model, authors estimated that the probability of false correspondence in at least 12 minutiae points between two arbitrary fingerprints each containing 36 minutiae, is 6×10^{-8}.

Later, Chen et al. [89] argued that Pankanti et al. made some simplifying and optimistic assumptions regarding minutiae properties, in particular regarding distribution of minutiae points as well as dependency between minutiae locations and their corresponding ridge orientations. In their work, they refined these assumptions and estimated more conservatively that the probability of false correspondence in at least 12 minutiae points between two arbitrary fingerprints each containing 36 minutiae, is 4.1×10^{-4}. Authors also demonstrate that, the probability distribution of impostor minutiae matching obtained using their model has a very good fit to the empirically obtained distribution, more so than any previous results.

Daugman developed one of the most successful iris recognition methods [20] using a 2048-bit representation of the iris (*iriscode*) which are simply matched using the Hamming distance. In order to assess iris entropy, he first empirically calculated the Hamming distance distributions of genuine and impostor irises by cross matching 1800 iris samples. Then, noting that a Binomial distribution with 173 bits of freedom fits very well to the above mentioned impostor distribution, Daugman concluded that each iris carries 173 bits of information (reduced from 2048 due to correlation between the bits in the IrisCode). In other words, his estimate for the probability of two unrelated irises to match is approximately 10^{-52} ($= 2^{-173}$).

Srihari et al. [82,83] aimed to empirically assess the individuality of handwriting as a biometric. To do so, they automatically extracted features from the grayscale images of handwritings and used the false accept and false reject rates as an estimate of individuality. Authors reported a 95% accuracy rate (or 5% false accept rate) for verification and 80% accuracy rate for identification of unknown document from a comprehensive database of 1000 different writers. However, as mentioned before FAR and FRR numbers do not extend beyond the scope of the particular database and matching algorithm used, thus falling short of truly estimating the individuality of the biometric modality.

On a more related study, Vielhauer et al. proposed a method for obtaining a *biometric hash* using online signatures [86, 87]. The method is based on adaptive quantization where extracted features are quantized into bins and the biometric hash is obtained by concatenation of the integer values representing each bin. Within this context, they analyzed the stability and entropy of 50 different features obtained from online signatures. To calculate stability, they measured the deviation of each feature within handwritings for a particular person and generalized across their test data set. For the entropy they calculated the average entropy of each feature. The highest feature entropy was calculated as 3.61 bits and only 17 other feature had entropy higher than 1.8 bits. Authors did not report total entropy of a signature.

Ballard et al. [92,93] implemented the system proposed by Vielhauer et al., with the aim of showing that a signature verification system could be compromised using synthetic forgeries generated using statistics obtained from the general population. Authors argued that the original approach proposed for the assessment of feature suitability did not represent susceptibility to forgery attacks. In other words, a feature may possess high entropy and stability, but may not be difficult to forge. For that reason, they analyzed 144 different features altogether and decided to use only 37 of these in the context of biometric hashing. While the original biometric hash implementation required exact match between all corresponding hash elements, Ballard et al. assumed that the biometric hash can be corrected if only a few features don't quantize into desired bins, either by search or by use of error correction

codes. They report 6.8% and 8.2% equal error rate (EER) when the system is tested using forgery signatures provided by untrained forgers who had access to offline and online renderings of the signatures, respectively. Finally, the system performed at 20.1% and 17.2% of EER when tested using forgeries provided by skilled forgers and synthetic signatures, respectively.

Feng and Wah proposed a private key generation method using online signatures [30]. The method is based on feature quantization and used only dynamic features of a signature. First, the range of each feature is calculated across all subjects to obtain database boundaries for that feature. During enrollment, user boundaries are found similarly and the database range for each feature is divided into bins of size equal to the user's range. Then, the indices of the bins where the user's features are mapped, are concatenated into a single vector from which the cryptographic hash value is calculated. In other words, quantization is done adaptively for each user. The hash value is then used to calculate a private key for that user. Authors report a performance of 8% equal error rate in generating the keys. They also analyze the entropy of each feature and conclude that online signatures contain on average 40 bits of entropy, calculated as the sum of individual feature entropies. Since the features may not be independent, this estimate of the signature entropy is an overestimate.

Brault and Plamandon attempted to quantitatively assess the complexity of a given signature [74], in terms of how easy it would be to forge it. The complexity calculation is based on the model of a human perception of a signature. It is assumed that a forger mentally/perceptually slices given signature according to its corresponding singular points (such as high curvature, etc.). So the difficulty of forging that signature depends on the cumulative complexity of reproducing each such slice, normalized by the signature's curvilinear length. The complexity of a given slice in turn depends on its curvilinear length and time required to reproduce that slice, as well as its average angle. This complexity coefficient was compared to forger's perception of difficulty, expert's opinion, and a dynamic time warping based distance measure between the forgery and forged signatures. The calculated complexity was

not strongly correlated with these measures, while the agreement between forgers' perception and the automatically calculated distance measure was strong. A similar study was performed by Elliott and Hunt [105] where they conducted a survey in order to assess perceptual difficulty of forging certain signatures. They showed a number of signatures to each forger and asked them to rate the difficulty of forging those signatures and classify them according to the signing speed. However, it was observed that the forgers' perception on both counts were significantly different from those of genuine owners of the signature.

As summarized above, the individuality or discriminative power of an online signature is mainly addressed at feature level, whereas more conclusive studies are done for fingerprint and iris modalities.

5.4 Proposed Signature Individuality Model

In a previous work [106], we explored the use of Fourier Descriptors for on-line signature verification. In particular we have showed that normalized Fourier Transform coefficients of a signature's y-profile (y-component of the signature's trajectory) could be used in online signature verification. The Fourier coefficients of a signature are global features which are attractive because one can represent a variable-length signature with a fixed number of features.

In section 5.4.1, we broadly describe this approach as it forms the basis of the proposed signature individuality model. A slight modification is made to the original matching algorithm in order to obtain an algorithm more amenable for comparison with the theoretical model. In the following sections we explaining feature extraction, matching, the proposed individuality model, the parameter estimation and obtained results (Sections 5.4.3-5.4.5).

5.4.1 Feature Extraction Using the Global Fourier Transform

An online signature can be represented as a complex signal in time dimension, specifying the x and y coordinates and optionally other features such as the pressure and time stamp, for each sampled point:

$$S(t) = \left[\begin{array}{cccc} x(t) & y(t) & pressure(t) & timestamp(t) \end{array} \right]^{\mathrm{T}} \qquad t = 1, 2, ..., N$$

where N is the number of points sampled along the signature's trajectory.

In this work, we apply the Fourier Transform to the y-profile of the signature which is composed of the y-coordinates of the sampled points (i.e. $y(t)$), discarding the additional information. The x-profile is discarded for simplicity because for signatures which are signed from left to right or right to left, the x-profile does not contain much discriminative information. Figure 5.1 depicts corresponding y and x profiles of three different signatures.

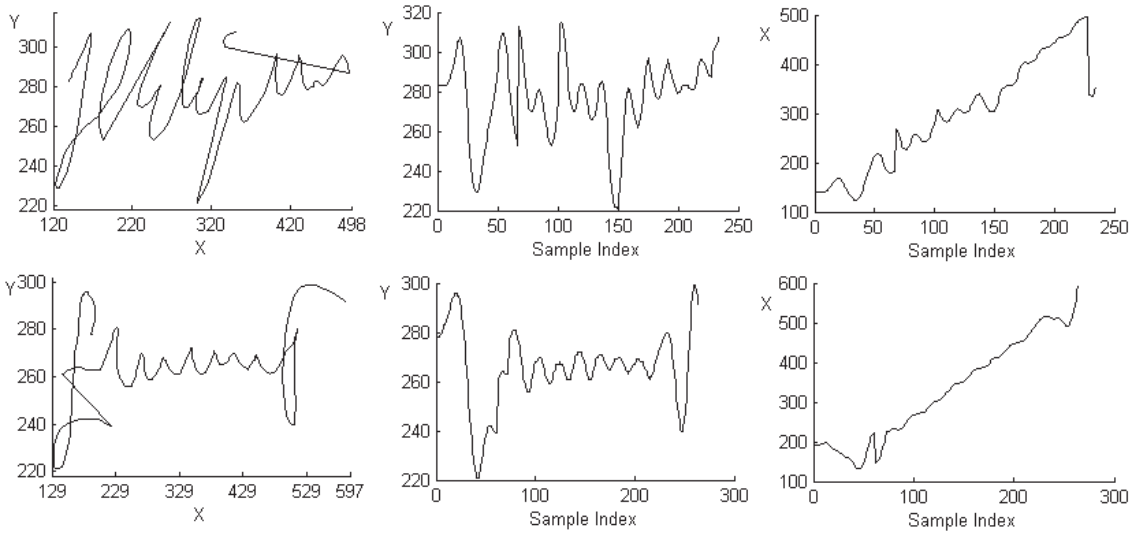

Figure 5.1: Two sample signatures (leftmost column) and their corresponding y (middle column) and x (rightmost column) coordinate profiles.

The Fourier Transform coefficients of a discrete signal $y(t)$ of length N is defined as follows:

$$C_k = \frac{1}{N}\sum_{t=0}^{N-1} y(t)e^{-j2\pi tk/N} \qquad k = 0, 1..., N-1 \tag{5.1}$$

In our case, $y(t)$ is the y-profile of the signature, N is the number of points in the signature and C_k is the k'th Fourier coefficient corresponding to the k'th harmonic. Given a complex coefficient $C_k = a_k + jb_k$, the magnitude of the coefficient, $|C_k| = \sqrt{a_k^2 + b_k^2}$ indicates the energy for the k'th harmonic, while $tan^{-1}(b_k/a_k)$ indicates its phase.

In 2D shape recognition, the Fourier Coefficients obtained by applying the Fourier Transform to the object's contour ($x(t)$,$y(t)$) can be normalized to achieve invariance against translation, rotation and scaling of the original shape [107]. For instance, translation of a shape corresponds to adding a constant term to each point of the original shape and affects (only) the first Fourier coefficient. By discarding C_0 and using the magnitudes of the remaining coefficients as features, we obtain invariance to translation (position of the signature on the tablet) and rotation(orientation relative to the tablet). Scale invariance is more complicated, due to the additional dimension of time. We have found that a robust approach to normalizing for scale variations is to divide each coefficient by the total amplitude of the Fourier Spectrum. Hence, our final features, also called the *Fourier Descriptors*, are obtained by normalizing the magnitudes of the Fourier coefficients C_k:

$$F_k = \frac{|C_k|}{\sum_{i=0}^{N-1}|C_i|} \qquad k = 1..., N-1 \tag{5.2}$$

Due to the natural variation in the signing process, genuine signatures of the same user almost never have equal lengths. The length variation results in Fourier Transforms with varying number of components; hence feature vectors of varying lengths. In order to obtain an equal number of Fourier Descriptors which then would correspond to the same frequencies, we pad each signature to be compared (reference set + query) with zeros, so as to match the length of the longest signature in the set, prior to the application of the Fourier Transform. Padding a signature

with zeros corresponds to padding a signal with zeros in the time domain.

5.4.2 Matching

During enrollment to the system, the user supplies a number of reference signatures which are used to measure the variation within his/her signatures, so as to set user-specific thresholds for accepting or rejecting a query signature. During verification, a subject provides his/her test signature to be compared against the claimed user's reference set signatures. After padding the signatures that are to be compared (i.e. reference and query) to the maximum length of the set, we apply the Fourier transform to the y-profile of each signature and calculate their corresponding Fourier Descriptors, as described in Section 5.4.1.

In order to match two signatures while allowing for some variability, we use quantization. For this, we first calculate the mean value (μ_k) for each Fourier Coefficient F_k over the reference set:

$$\mu_k = \frac{1}{R}\sum_{i=1}^{R} F_k^i \qquad k = 1..., n \tag{5.3}$$

where F_k^i denotes the k'th Fourier coefficient of the i'th reference signature, R denotes the number of reference signatures, and n denotes the fixed number of Fourier coefficients calculated for each signature. We then calculate the range of each Fourier descriptor F_k, by finding the maximum and minimum values of F_k over the whole database. Next, we adaptively quantize this range into a constant number (t) of bins such that the mean reference value (μ_k) is at the center of its corresponding bin. The bin widths are thus determined so that each coefficient's range is divided into exactly t bins (extending the range slightly when necessary to fit t bins). The Fourier Descriptors can take on any values in [0-1] due to the normalization procedure we use; but in practice they range in [0-0.3].

Finally, the query signature is accepted if at least m_0 out of its n Fourier Descriptors fall into the same bin with the corresponding mean of the reference set (i.e. μ_k). Figure 5.2 illustrates the matching process, showing a case of match for

the first harmonic (both query coefficient and the corresponding reference mean fall into the same bin) and a case of mismatch for the second harmonic.

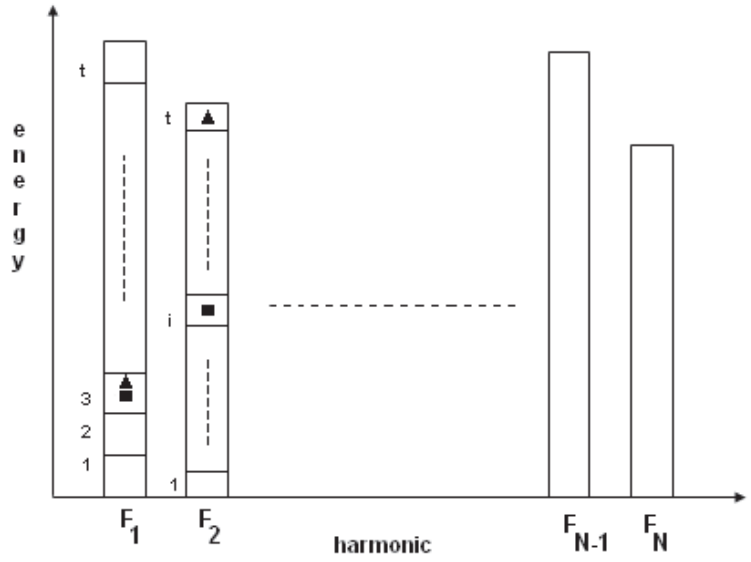

Figure 5.2: A matching illustration of a query signature to a reference set. The range of each harmonic (F_i) is divided into a constant number of bins (t). Query signature's descriptor (triangle) is said to match its corresponding reference set's mean (circle) if they both fall into the same bin, as is the case for F_1 but not F_2.

The feature extraction and matching algorithms presented here are tested using the publicly available SUSIG database which contains 1000 enrollment (5 for each subject) and 3940 test (total of genuine and forgery) signatures provided by 200 subjects [108]. Using a total of 25 feature values (n), and requiring a match in 13 or more feature values (m_0) as dictated by EER performance, we obtained a 14% equal error rate for random forgeries. During the test, we divided each feature value's range into 8 bins; the selection of the number of bins is explained in Section 5.4.4. While these results are inferior to the state-of-the-art (around 3% EER) [98], it is comparable to those achieved by top-10 systems participated in the Signature Verification Competition (SVC2004) [104], indicating the potential of Fourier Descriptors as successful features for online signature verification.

In fact, the above mentioned matching algorithm is slightly modified from another signature verification system using Fourier descriptors which has a much lower equal error rate (7.5%) on the same database [106]. The modification is done so that the similarity of two signatures is based on the total number of matching harmonics, rather than total distance to the mean features, allowing for an easier theoretical calculation of the entropy. The reason we have chosen the Fourier domain representation is the fact that global features are more amenable for the individuality analysis compared to local features, as explained in the following section.

5.4.3 The Individuality Model

Online signature doesn't have a de facto representation: different automated signature verification systems use different feature sets. This is in contrast to fingerprints, for which minutiae representation is accepted both by human experts and most of the automated verification systems.

Our signature individuality model is based on the Fourier domain representation of the signature's trajectory as described in the previous section 5.4.1. The Fourier domain representation is suitable for assessing the individuality of signature as a biometric since, first of all, because the Fourier Transform coefficients represent the original signal in a lossless fashion, yet, one can remove the high frequency components to achieve a more compact and fixed-length representation. How many components can be removed without affecting the characteristic features is not clear, however due to our previous studies on signature verification, we are using the first 25 components in the current model. Figure 5.4 demonstrates original and corresponding reconstructed y-profiles of four different signatures. Local features, such as velocity or relative angle at each point on signature's trajectory are in general more useful for verification, as they capture local shape and timing variations [98]; however global features make the task of matching two signatures easier (e.g. typically a simple Euclidian distance is used). Assessing individuality is also easier with global features since the number of features is fixed. In contrast, with local features, the number of features depend on the length of the signature which typically shows

large variations. As a result, matching algorithms (e.g. dynamic programming) and costs are highly non-linear, making the analysis a challenging problem.

We base our individuality work on that of original work by Pankanti et al., which is later improved by Chen et al. Similar to their work, we formulate the individuality problem as the probability of coincidentally matching a given arbitrary signature. Using the Fourier domain representation, we first look at the probability of match in a single harmonic: two signatures are said to match in a given harmonic if their Fourier descriptors (normalized Fourier coefficient magnitude) for that harmonic are suitably close. How to judge the proximity of two harmonics given inter and intra-class variations is the main question here. Similar to what is done in Pankanti et al., we find the amplitude threshold d_0 such that a high percentage of corresponding genuine coefficients are considered as matching.

Next, we consider the probability of match for a m out of n of their corresponding harmonics. That probability can be modeled using the binomial distribution, which is formulated as follows:

$$P(m|n) = \begin{pmatrix} n \\ m \end{pmatrix} p^m (1-p)^{n-m} \tag{5.4}$$

where p is the probability of match in one harmonic. Here we make the assumption that the probability p of match for each harmonic is equal. The binomial distribution provides the discrete probability distribution $P(m|n)$ of obtaining exactly m successes in a sequence of n independent binary events, also called Bernoulli trials, where the outcome of each trial is either success or failure with corresponding probabilities of p and $1-p$, respectively. In the context of our model, the success corresponds to two corresponding coefficients falling into the same bin. Since two signatures are considered as matching in the case where at least m out of n harmonics match, the probability of match between two signatures (P_{match}) is the sum

of probabilities of all such cases:

$$P_{match} = \sum_{m=m_0}^{n} P(m|n) \tag{5.5}$$

where m_0 is the minimum accepted number of matching descriptors and n is the total number of descriptors.

For the time being we assume that the coefficients are uniformly distributed within the range of the harmonic (i.e. the probability for a coefficient to fall into a bin is equal for all the bins). In fact, the deviation from uniform distribution is not that severe and the necessary update for the analysis will be done in the future. Another shortcoming of the proposed approach may be about the assumption of independence of the Fourier coefficients of online signatures. One may argue that these are not independent, thus the matching of coefficients are not independent events. In that case, the entropy estimate would need to be lowered to take into account the number of dependent dimensions. While Figure 5.3 and Table 5.4.3 show that the Fourier descriptors corresponding to consecutive frequencies do indeed have low correlations, the rank analysis of the descriptor matrix show a full rank, indicating that there is no clear linear dependence between the descriptors. Further studies would need to be done in order to measure and/or remove more complex or subtle dependencies, or to extend the model to account for them.

So far we have presented a model depending on a number of parameters which must be estimated in order to assess the signature's individuality. In the next section we estimate these parameters using a large database.

5.4.4 Parameter Estimation

The proposed model is governed by a number of parameters such as i) the number of harmonics used to represent a signature (n), ii) the probability of coincidentally matching a single harmonic (p), and iii) the minimum number of matching harmonics required to consider two signatures as similar (m_0). In this section, we discuss the parameter estimation of the model and assess the individuality of the online

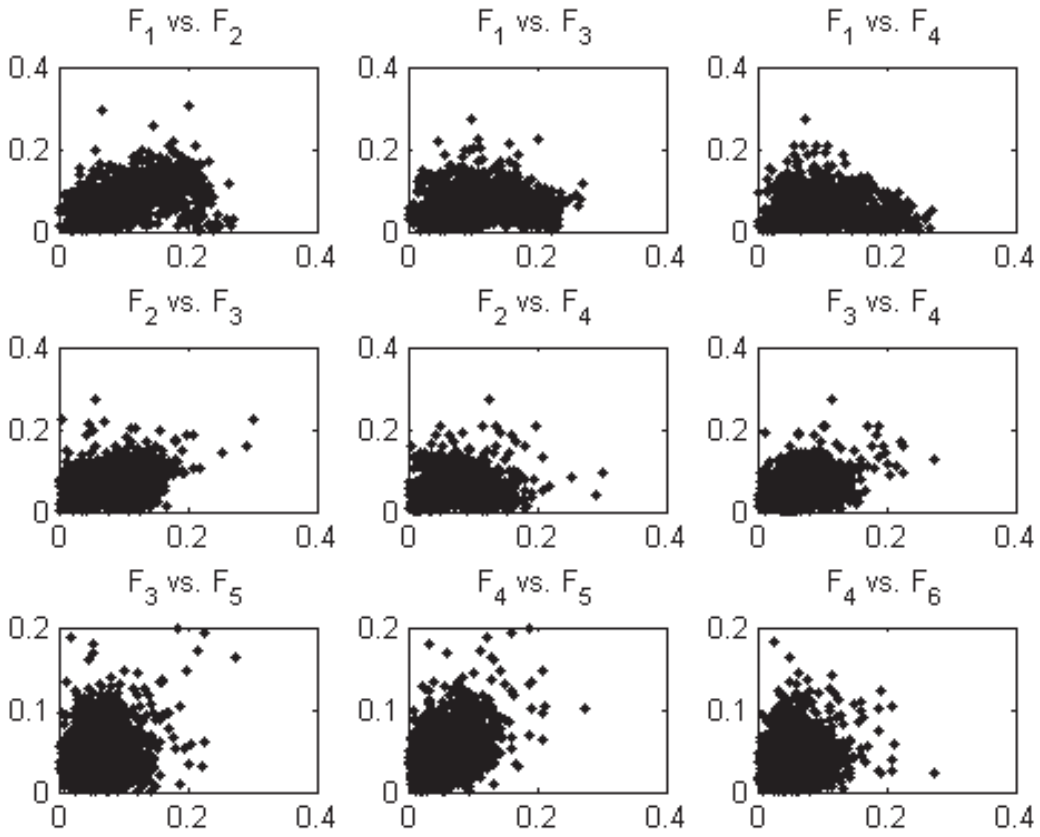

Figure 5.3: Pairwise distribution of some of the Fourier descriptors, calculated using the SUSIG database.

signature.

All of the parameter estimations are carried out using a publicly available online signature database SUSIG [108]. The database was collected using genuine signatures provided by 200 subjects for a total of 3940 genuine and 2000 skilled or highly skilled forgeries. However, since we are interested in the guessing entropy, we have only used the random forgery protocol of the database (297,000 random forgeries). Whenever the system performance is assessed, we set apart 5 reference signatures for each subject and the rest is used as the corresponding test set.

To estimate p, we looked at the distribution of the distances of Fourier descriptors over the genuine population; we then selected a match threshold d_0 such that 90% of the genuine pairs are considered as matching. Since different descriptors may have

	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
F_1	1.00	0.50	0.16	-0.12	-0.13	-0.12	-0.19	-0.26	-0.20	-0.24
F_2	**0.50**	1.00	0.41	0.08	-0.06	-0.06	-0.15	-0.19	-0.18	-0.22
F_3	0.16	**0.41**	1.00	0.38	0.19	0.01	-0.04	-0.12	-0.11	-0.14
F_4	-0.12	0.08	**0.38**	1.00	0.42	0.16	0.05	0.04	-0.01	-0.08
F_5	-0.13	-0.06	0.19	**0.42**	1.00	0.27	0.20	0.07	0.07	0.03
F_6	-0.12	-0.06	0.01	0.16	**0.27**	1.00	0.29	0.16	0.03	0.01
F_7	-0.19	-0.15	-0.04	0.05	0.20	**0.29**	1.00	0.37	0.15	0.11
F_8	-0.26	-0.19	-0.12	0.04	0.07	0.16	**0.37**	1.00	0.31	0.22
F_9	-0.20	-0.18	-0.11	-0.01	0.07	0.03	0.15	**0.31**	1.00	0.30
F_{10}	-0.24	-0.22	-0.14	-0.08	0.03	0.01	0.11	0.22	**0.30**	1.00

Table 5.1: Correlation matrix for first 10 Fourier descriptors, calculated using the SUSIG database.

different ranges, we used normalized distances in this process (distances divided by the range of the harmonic). We found d_0 to be 0.126. As a result, we quantize each descriptor's range by bins of width $d_0 = 0.126$; hence using 8 ($= 1/0.126$) bins. Since we consider two descriptors as matching if they fall within the same bin and since we use normalized distances, we set $p = d_0$.

Given p, we have two other parameters to estimate, namely the number of harmonics required to represent a signature (i.e. n) and the number of required matching harmonics (i.e. m_0). To do this, we exhaustively searched over all possible n values, where for a fixed n, we determined m_0 as the number of matching coefficients giving equal error rate (EER). We obtained best results for $n = 25$, with corresponding values of $m_0 = 13$ and $EER = 14\%$. Notice that $n = 25$ is sufficient for online signatures, as indicated by low reconstruction errors shown in Figure 5.4.

5.4.5 Results

Once these estimated parameters are plugged in to our model, the probability of a coincidental match (P_{match}) between two signatures is calculated as 2.4×10^{-6}, which corresponds to an entropy of approximately 19 bits. This estimate is somewhat optimistic, since the uniform distribution assumption within a given harmonic is

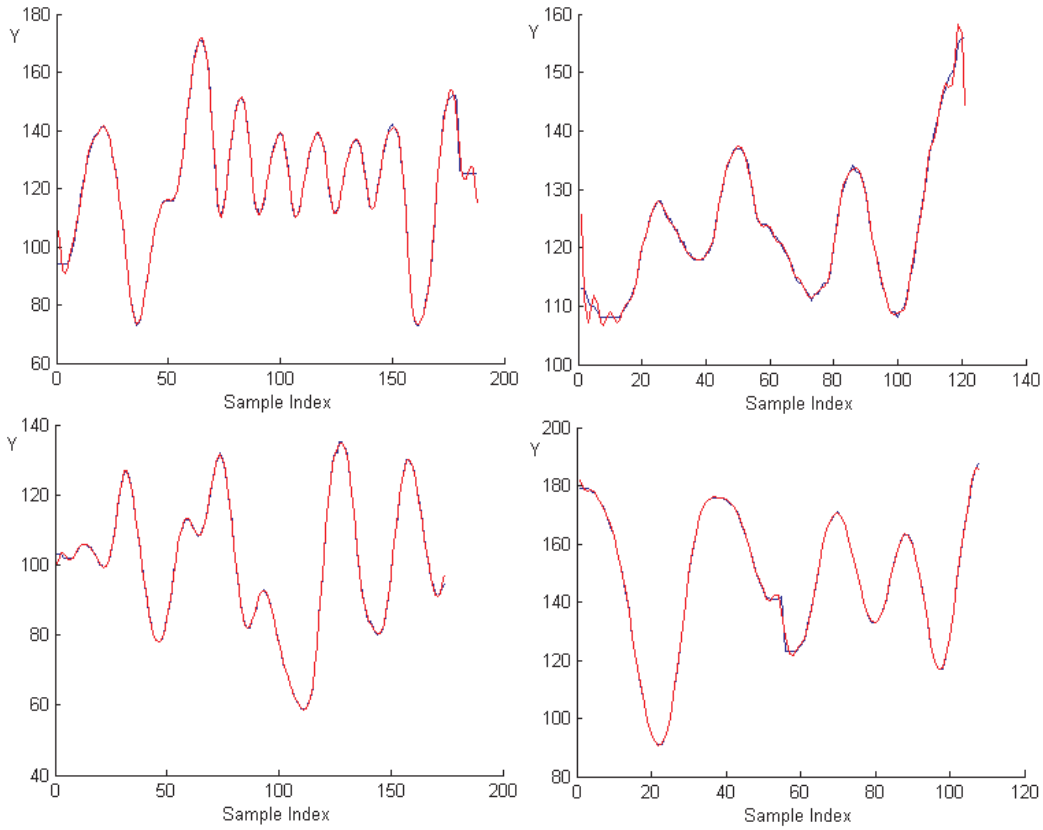

Figure 5.4: The original 4 y-profiles (red) overlapped with their corresponding reconstructed versions (blue). The reconstruction is done using the inverse Fourier transform of the first 25 Fourier coefficients.

actually not valid, but used for simplification. We have in fact calculated that the average entropy of a single descriptor is about 2.3 bits, which corresponds to a probability of a single match (p) of approximately 0.2 ($1/2^{2.3}$), assuming uniform distribution. When we plug this new probability into the entropy estimate for the whole signature, the probability of a coincidental match becomes 4.7×10^{-4}, which corresponds to a more conservative estimate of 11 bits.

Figure 5.5 demonstrates impostor and genuine probability distributions estimated using our model (A and B) and empirically calculated distributions (C and D) using the matching algorithm explained in the Section 5.4.2 over the SUSIG database. Specifically, the distributions labeled by A and B depict impostor dis-

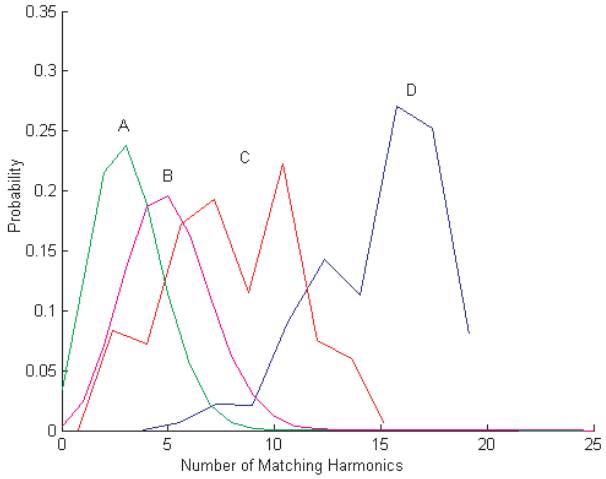

Figure 5.5: Distributions labeled by A and B depict the theoretical estimates for number of coincidental matches between two signatures using $n = 25$, $k = 13$, while p set to 0.126 and 0.2, respectively. Distributions labeled by C and D depict impostor and genuine distributions obtained from the SUSIG database using the same parameters.

tributions estimated using our model with the probability of match p set to 0.126 and 0.2, respectively. The distributions labeled by C and D represent empirically calculated impostor and genuine probability distributions obtained using the same parameters ($n = 25$, $m_0 = 13$, $p = 0.126$), respectively. The empirically calculated impostor distribution (C) is close to that estimated using our model, but shifted apart by a few coefficients, which we believe is due to the simplifying assumptions made in deriving the model; in particular, the uniform distribution of the descriptors among the bins.

5.5 Summary

The uniqueness of a biometric modality determines its discriminative power, as well as providing insights into its performance when deployed in a large-scale application.

In this work, we proposed a preliminary individuality model for online signatures, where the individuality of the signature is estimated by the probability of coincidental match between two different signatures. We used the Fourier domain

representation of the signature to simplify the theoretical analysis. We have estimated the model's parameters using a large database of online signature and showed that an average signature possesses between 11-19 bits of entropy, when we make certain simplifying assumptions. This entropy values correspond to an accidental match probability of 4.7×10^{-4} to 2.4×10^{-6}. We also show that the impostor probability distribution estimated by our model is close to that calculated empirically.

This is a preliminary work in that some simplifying assumptions are made. These assumptions affect how the theoretical model (i.e. A and B in Figure 5.5) matches the empirical model (i.e. C) and addresses how good the model is. On the other hand, there is also the issue that the Fourier domain representation is not a standard representation for online signatures and the accompanying matcher that is used in deriving the theoretical entropy formula has a relatively high error rate (i.e. the overlap between C and D, in Figure 5.5). This, in turn, makes the individuality model less useful/informative since extrapolating system performance for very large databases is meaningful when the error is very small. As an example, consider the case where a system achieves 0% EER on a 1000-people database; in that case, using theoretical models to claim that the performance with very large databases would be similar, is very important. Nonetheless, the results are significant given that this is the first individuality model for online signatures, while previous work mostly dealt with feature level discriminative capabilities. In fact, while online signatures are often thought as less secure compared to fingerprints and irises, this work highlights that online signatures already possess a relatively high level of entropy and as a behavioral biometric with adjustable complexity, it stands as an important biometric modality.

Chapter 6

SUSIG: Online Signature Database

We present a new online signature database (SUSIG). The database consists of two parts that are collected using different pressure sensitive tablets (one with and the other without an LCD display). A total of 100 people contributed to each part, resulting in a database of more than 3000 genuine signatures and 2000 skilled forgeries. The genuine signatures in the database are real signatures of the contributors. In collecting skilled forgeries, forgers were shown the signing process on the monitor and were given a chance to practice. Furthermore, for a subset of the forgeries (highly skilled forgeries), this animation was mapped onto the LCD screen of the tablet so that the forgers could *trace over* the mapped signature. Forgers in this group were also informed of how close they were to the reference signature, so that they could improve their forgery quality.

We describe the signature acquisition process and several verification protocols for this database. We also report the performance of a state-of-the-art signature verification system using the associated protocols. The results show that the highly skilled forgery set is significantly more difficult compared to the skilled forgery set, providing researchers with challenging forgeries. The database is available through http://biometrics.sabanciuniv.edu.

6.1 Introduction

In a signature verification system, users are first enrolled by providing signature samples, called *reference signatures*. Then, when a user presents a *test signature* claiming to be a particular individual, this test signature is compared with the reference signatures of the claimed identity. If the resulting dissimilarity is below a certain threshold, the user is said to be verified. In evaluating the performance of a signature verification system, there are two important factors: the false rejection rate (FRR) of genuine signatures and the false acceptance rate (FAR) of forgery signatures. Since the two error rates are inversely related, a commonly reported performance measure is the Receiver Operating Characteristic (ROC) curve which shows how true accept rate (1-FRR) changes with FAR, for different acceptance thresholds. When only a single performance measure is required, for instance while comparing different systems, the equal error rate (EER) that denotes the point on the ROC curve where FAR equals FRR, is often reported.

Since FAR and FRR rates depend on the particular database used in testing a system, it is important that the database is as realistic as possible. For instance, a database containing genuine signatures collected in a single session would give an optimistic estimate of the true error rates (FAR and FRR that would be observed if the system was deployed in a real application). This is because in real applications, genuine test signatures are signed over an extended period and thus show higher dissimilarity to reference signatures in general, compared to those collected along with the reference set. On the other side, obtaining real forgery signatures is impractical, if not impossible, since it requires forgers/criminals who are motivated to break into the system. Instead, two forgery types have been defined in literature: a *skilled* forgery is signed by a person who has had access to a genuine signature for practice, and a *random* forgery is signed without having any information about the signature to be forged. Since a so-called skilled forgery may not actually be skilled at all, sometimes the term *informed* forgeries is used to denote those forgeries where forgers see the signature they are asked to forge, but not necessarily practice enough

to be called skilled. We use the term skilled forgeries in this database since it better represents the efforts and resulting skills acquired by forgers that were asked to practice before forging.

6.2 Previous Work

Publicly available databases and associated verification protocols make it possible to objectively compare different verification algorithms. We are aware of 3 large public databases containing online signatures: the MCYT database [109], the BIOMET database [110], and the SVC2004 database collected for the First International Signature Verification Competition [104].

The MCYT database [109] is a large multimodal database containing fingerprint and handwritten signatures. There are a total of 16500 signatures collected from 330 individuals (a 100-person subpart is made public), with 25 genuine and 25 forgery signatures collected for each person. The forgeries for each person are provided by 5 other individuals from the database. The signatures are obtained using a Wacom Intuos tablet with an Ink Pen, providing the x and y-coordinates, pressure, azimuth and altitude for each point on the trajectory. One shortcoming of this database is that genuine signatures (reference and test) are obtained in a single session. As mentioned before, signatures collected in a single session are expected to be more similar, compared to those collected with time elapse. Another important issue in a biometric database is the forgery quality. In the MCYT database, forgers observed only static images of signatures they were forging. While this may be the more realistic case, it underestimates the forgery quality levels achievable by genuine impostors who may obtain dynamic information about the signature. Finally, for this database, timestamps are not provided, so the pen-up durations cannot be estimated with certainty.

The BIOMET database [110] is also a large multimodal database which contains data from 5 different modalities: fingerprint, face, voice, hand and on-line handwritten signature. The database is collected in 3 different campaigns, with 130, 106

and 91 people. There are 15 genuine signatures and 17 forgery signatures collected for each person. The forgeries for each person are provided by 5 other individuals from the database. The signatures are obtained with a Wacom Intuos2 tablet, providing the x and y-coordinates, pressure, azimuth and altitude for each point on the trajectory, however different pens were used for the first and last two sessions: Wacom's Grip Pen (no visual feedback) was used for the first session and Wacom's Ink Pen was used for last two sessions. Using the Ink Pen, subjects could naturally sign their signatures on an ordinary sheet of paper placed over the tablet. The forgery signature acquisition process is not explicitly mentioned and there are no verification protocols associated with this database.

Finally, SVC2004 database [104] is a large database containing signatures from 100 individuals, with 20 genuine and 20 forgery signatures collected for each person, amounting to a total of 4000 signatures. Genuine signatures are collected in two different sessions. Forgeries for each person are provided by at least 4 other individuals from the database. The signatures are obtained with a Wacom Intuos tablet. The main strengths of this database are its size and the availability of benchmark performance results of various signature verification systems that participated in the SVC2004 competition. On the other hand, a shortcoming of this database is the fact that people have not used their real signatures, but instead provided made-up signatures which they practiced solely for the sake of contributing to the database. As real signatures are signed through a ballistic motion, made-up signatures would be expected to have higher variance and consequently cause higher error rates. In fact, while the average EER results for skilled forgeries reported by SVC2004 is not high (2.8%), there is a high variation between the EER results of randomized tests using different genuine signatures as reference set.

6.3 SUSIG Database

Hereafter, we describe a new online signature database, called SUSIG (Sabanci University Signature database) and associated verification protocols. The database

aims to address some of the shortcomings of the previous databases and provide one more public benchmark database to serve the research community. In the following sections, we describe in detail the database construction as well as its associated protocols.

The SUSIG database consists of two subcorpora: Visual and Blind. Signatures in the Visual subcorpus were collected using a pressure sensitive tablet with built-in LCD display such that people could see their signatures while signing, whereas no visual feedback was available for the Blind subcorpus. The Blind subcorpus was collected approximately 4 years before the Visual subcorpus; as a result, the people who donated to the two subcorpora are mostly different but share similar demographics, resulting in similar signature complexities.

The signatures in the database are real signatures of the participants, which often, but not always, include names or abbreviated names of the participants written in the Latin alphabet. Samples of the signatures from the Visual and Blind subcorpora are shown in Figures 6.1 and 6.2.

Figure 6.1: Sample genuine signatures from the SUSIG Visual Subcorpus.

Figure 6.2: Sample genuine signatures from the SUSIG Blind Subcorpus.

6.4 Signature Acquisition

The signature acquisition hardware that is available in the market can be categorized into two major groups: i) smart pens and ii) pressure sensitive tablets. Smart pens generally have force sensors on the pen tip, sensing the movement of the pen and acquiring the signature trajectory while the pen is moving. On the other hand, pressure sensitive tablets perceive the pressure exerted by the pen tip on the tablet. Pressure sensitive tablets themselves can be divided into two groups: those with visual feedback provided through an LCD display on the tablet and others without one. Tablets with visual feedback are more comfortable because people can see what they are signing, while tablets without such capability are cheaper.

Depending on the hardware used, the following features are commonly measured at each sample point on a signature trajectory: i) x and y coordinates of the pen tip, ii) pressure exerted by the pen, iii) time stamp, iv) azimuth of the pen ($0 - 360^o$), v) altitude of the pen ($0 - 90^o$) with respect to the signing surface. Using these measured features, system developers may extract many other features (such as velocity, acceleration, etc.) as required by their algorithms [94, 95, 103, 111, 112].

For the Blind subcorpus, we used Wacom's Graphire2 pressure sensitive tablet and pen. The tablet's active area is 5.02x3.65 inches with 1000 lines per inch spatial resolution. The tablet has a sampling rate of 100Hz, recording at each sample point the x and y coordinates of the signature's trajectory, pressure (512 levels) and the

time stamp. Wacom's pen is featured to capture samples only during the interaction of the pen tip with the tablet (pen-up times can be identified by the time-stamp difference).

For the Visual subcorpus, we used Interlink Electronics's ePad-ink tablet which has a pressure sensitive LCD screen. The LCD screen dimensions are 3x2.20 inches with a 300dpi spatial resolution. The tablet has a sampling rate of 100Hz, recording at each sample point the x and y coordinates of the signature's trajectory, pressure (128 levels) and the time stamp.

Each signature in the database is saved as a text file, containing the x and y coordinates, time stamp, and pressure level for each point on the signature trajectory.

6.5 Signature Animation Tool

To collect skilled forgeries (for either subcorpus), we added a simulation module that animates the signing process of a given signature on the monitor, so that forgers could see the signature trajectory and the signing speed. Specifically, the animation tool draws the sample points of the signature in order, ignoring the pen-up durations. In the case of a complex signature where the signature's trajectory is not obvious from its image, this animation provides valuable information to the forger.

In order to collect highly skilled forgeries for the Visual subcorpus, we mapped this animation onto the built-in LCD display of the tablet and let forgers practice and forge the signature, by *tracing over* the mapped reference signature. While the static image of a signature is much more likely to be compromised, we aimed to push the limits of the forgery quality so as to simulate a scenario where dynamic information about a signature is compromised.

6.6 The Visual Subcorpus

The Visual subcorpus consists of signatures donated by 100 people (29 women and 71 men). Most of the subjects were students or faculty members of Sabanci Uni-

versity, with ages varying between 21 and 52. Each person was briefly informed of the purpose of the data collection, without further information about the working principles of an online signature verification system.

The Visual subcorpus was collected in two separate sessions (VS1 and VS2) that were approximately one week apart. Each person supplied 10 samples of his/her regular signature in each session, for a total of 20 genuine signatures, without any constraints on how to sign. Each person was then asked to forge a randomly selected user's signature. The forger had a chance to watch the signature's animation several times on the monitor and practice it a couple of times before forging. Forgers were always shown the same reference signature (the first signature of session VS1) of the selected user, so that they could improve over time. During the practice, the forgers were also able to see other reference signatures of that user, to understand the variation among the reference signatures, though this was only used by some forgers. When they were satisfied with their skill level, forgers provided 5 forgeries of the signature they were asked to forge. For each subject in the Visual subcorpus, we thus collected 5 skilled forgeries by someone else from the same corpus.

In collecting what we call highly skilled forgeries, the animation of the reference signature was not only shown on the monitor, but also mapped onto the LCD screen of the tablet so that the forgers could trace over, as explained in Section 6.5. For each subject in the Visual subcorpus, we collected 5 highly skilled forgeries provided by the same two people. In order to obtain better quality signatures, forgers in this

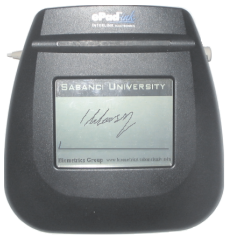

Figure 6.3: Signature animation done on the built-in tablet used in the Visual Subcorpus.

group were also informed of how close their forgery was to the reference signatures of the claimed identity, as feedback. Finally, again for this group, forgeries that were visibly dissimilar were discarded, to better simulate a true imposter. The similarity between the forgery signature and the corresponding reference signatures was measured with the metric used in our benchmark system [98]. This metric only uses the signature trajectory and the order of the sample points (pressure and time stamps are not used). Hence, while the feedback depends on the specific distance metric of the benchmark system, we think that the quality improvement in subsequent forgeries is absolute since the metric is quite successful in separating forgeries from genuine signatures.

In summary, 20 genuine signatures and 5 skilled and 5 highly skilled forgeries were collected for each person in the subcorpus (forming the subsets VS1, VS2, VSF and VHSF, respectively). Additionally, we have a separate 10-person validation set (VV) with 10 genuine and 10 forgery signatures per person acquired in a single session, that can be used to tune system parameters. Table 6.1 summarizes the Visual subcorpus.

Data Set	**Type**	**Users**	**Samples/User**	**Size**
VS1	Genuine	100	10	1000
VS2	Genuine	100	10	1000
VSF	Skilled Forgery	100	5	500
VHSF	Highly Skilled Forgery	100	5	500
VV	Genuine/Forgery	10	10/10	200

Table 6.1: Summary of the SUSIG Visual Subcorpus. The first 4 rows refer to the same 100 people, but the signature samples in each row are mutually exclusive.

6.7 The Blind Subcorpus

The Blind subcorpus was collected approximately 4 years before the Visual subcorpus and is named as such since the collection was done on a tablet without visual feedback. The subcorpus consists of signatures donated by 100 individuals (25 women and 65 men), most of whom were students and faculty members of Sabanci University, with ages varying between 20 and 50. Each person was briefly informed of the purpose of the data collection without further information about the working principles of an online signature verification system.

Each person was first asked to supply samples of their regular signature on the provided pressure sensitive tablet, without any constraints on how to sign. First group of 30 people provided 8 genuine signatures, while the rest of the 70 people supplied 10 genuine signatures each. People in either group supplied their signatures in a single session.

Each person was then asked to forge a randomly selected user's signature. The forger was given the first reference signature of the selected user; he/she then had a chance to watch the signature's animation several times and practiced it a couple of times before forging it. When they were satisfied with their skill level, forgers provided 10 forgeries of the signature they were asked to forge. For each subject in the Blind subcorpus, we thus collected 10 skilled forgeries by one other person (BSF).

In summary, 8 or 10 genuine signatures and a total of 10 skilled forgeries were collected for each person in the subcorpus (forming the subsets BS1 and BSF, respectively). Additionally, we have a separate 10-person validation set (BV) with 10 genuine and 10 forgery signatures per person acquired in a single session, that can be used to tune system parameters. Table 6.7 summarizes the Blind subcorpus.

Data Set	Type	Users	Samples/User	Size
BS1	Genuine	100	8/10	940
BSF	Skilled Forgery	100	10	1000
BV	Genuine/Forgery	10	10/10	200

Table 6.2: Summary of the SUSIG Blind Subcorpus. The first 2 rows refer to the same 100 people, but the signature samples in each row are mutually exclusive.

6.8 Verification Protocols

In this section, we define verification protocols which should be used while assessing and reporting system performance results using the SUSIG Database.

A verification protocol is a set of rules which must be followed while assessing the performance of a system using a certain database. In particular, a protocol defines which data should be used for enrolling users to a system (reference set), tuning system specific parameters (validation set) and testing the performance (test set). In order not to bias performance results, people who provide signatures to the validation set must not provide any signature to the reference or test sets. Similarly, signatures from the reference set of a person must not be included in the test set of that person. It is also essential for an unbiased performance assessment that all system-wide parameters, in particular those affecting the calculation of similarity scores, are kept fixed once they are trained using the validation set. The only exception to this is the threshold used to accept or reject a signature given its similarity score(s). This threshold may be decreased or increased to reduce FRR or FAR respectively, so as to find the EER.

The SUSIG verification protocols are designed to test a system's performance under various conditions. In the following sections, we define 7 such protocols. Note that some of the protocols are designed for the sake of completeness and compatibility with previous databases (e.g. the single session protocol), while others correspond to more realistic scenarios (e.g. across-session protocol). Each protocol is applicable to either one of the subcorpora, unless explicitly stated otherwise.

Each protocol uses signatures from one subcorpus, except for the device-independent protocol. Details of the protocols can be found in Table 6.8, while descriptions and further explanations are given in sections 6.8 through 6.8.

We have envisioned 5 genuine signatures to be used as reference signatures, as is the case with the SVC2004 database. However, if a system requires more than 5 reference signatures, most of the protocols may be modified appropriately, adding more of the genuine test signatures to the reference set.

The SUSIG protocols are designed to be used in an open-set verification format where new users can be added to or deleted from the system without the need of re-configuring/re-tuning the system and its verification specific parameters.

Protocol	**Reference**	**Genuine Test**	**Forgery Test**
Visual SubCorpus			
Single-Session	VS1[1..5]	VS1[6..10]	VSF+VHSF
	VS2[1..5]	VS2[6..10]	VSF+VHSF
Across-Session	VS1[1..5]	VS2[1..10]	VSF+VHSF
	VS2[1..5]	VS1[1..10]	VSF+VHSF
Base (=MS)	VS1[1..5]	VS1[6..10]+VS2[1..10]	VSF+VHSF
Skilled Forgery	VS1[1..5]	VS1[6..10]+VS2[1..10]	VSF
Highly Skilled Forgery	VS1[1..5]	VS1[6..10]+VS2[1..10]	VHSF
Random Forgery	VS1[1..5]	VS1[6..10]+VS2[1..10]	Others' genuine sigs.
Blind SubCorpus			
Base(=SS/SF)	BS1[1..5]	BS1[6..8/10]	BSF
Random Forgery	BS1[1..5]	BS1[6..8/10]	Others' genuine sigs.
Whole Database			
Device Independent	VS1[1..5]	BS1[1..8/10]	BSF
	BS1[1..5]	VS1[1..10]+VS2[1..10]	VSF+VHSF

Table 6.3: Summary of the Protocols. VS1,VS2,VSF, VHSF, BS1, and BSF refer to the subsets defined in subsections 6.6 and 6.7. SS, MS, SF refer to Skilled Session, Mixed Session and Skilled Forgery, respectively. The forgeries in each experiment are obtained from the corresponding subcorpus only, except for the Whole Database protocols. The protocols marked in **bold** are the essential protocols, while the others measure performance under certain restricted conditions.

Single-Session Protocol

This protocol assesses the performance of a system under the condition where both reference and test data are obtained in the same session. The protocol proceeds as follows: the first 5 genuine signatures of each user are used as reference and the remaining signatures of the *same* session are used as test signatures. All available forgery signatures of the subcorpus are also added to the test set.

For the Visual subcorpus, another test should be run using the signatures in the second session. Finally, all of the verification results (genuine and forgery scores obtained in the two experiments) should be merged and the EER should be calculated over the combined results, using a *single* threshold. This is preferred to averaging the EER results separately calculated for each experiment, since the reference and genuine sets of the two experiments are fully disjoint. Note that while the forgery signatures used in the two experiments are the same, forgery results are not the same, as the reference sets are different.

This protocol is the base protocol for the Blind subcorpus.

Across-Session Protocol

The goal of this protocol is to assess the performance of a system under the more realistic conditions where genuine signatures for reference and test sets are obtained in different time periods. This protocol is applicable only to the Visual subcorpus where genuine signatures are obtained in two different sessions.

The protocol proceeds as follows: the first 5 genuine signatures of one session are used as reference and all signatures of the *other* session, as well as all forgery signatures, are used as test signatures. After this test, roles of the two sessions must be interchanged and the EER should be calculated over the combined results of the two experiments, using a *single* acceptance threshold.

Mixed-Session Protocol

The goal of this protocol is to use all available genuine signatures in the Visual subcorpus; hence, it is a mixture of the single and across-session protocols.

The protocol proceeds as follows: first 5 genuine signatures of the first session are used as reference and all other genuine signatures (i.e. rest of the first session and all of the second session) are used as test signatures. All available forgery signatures are also used as test set. Note that since all genuine data is already used, we do not run a second experiment where the roles of the two sessions are interchanged.

This protocol is the base protocol for the Visual subcorpus.

Skilled Forgery Protocol

The goal of this protocol is to assess the performance of a system against *skilled* forgeries only, discarding highly skilled forgeries. This protocol is applicable to either subcorpus; however, as there is no highly skilled forgeries in the Blind Subcorpus, this protocol is the same as the single session (Base) protocol for the Blind Subcorpus.

Highly Skilled Forgery Protocol

The goal of this protocol is to assess system performance against *highly skilled* forgeries, only. The protocol is applicable only to the Visual subcorpus, as the Blind subcorpus doesn't contain highly skilled forgeries. This protocol is the same as the base protocol of to the Visual subcorpus, except for the fact that skilled forgeries are excluded from the test set.

Random Forgery Protocol

The goal of this protocol is to assess system's performance against a random forgery attack. In a random forgery attack, forgers don't have any prior information regarding the signature to be forged; so genuine signatures of all other users are used as random forgeries for each user.

To perform the random forgery test, first 5 genuine signatures of the first session are used as reference, while the rest of the signatures are used as genuine test signatures and all the genuine signatures of all other subjects are used as forgery test signatures. This protocol is applicable to both subcorpora.

Device-Independent Protocol

The goal of this protocol is to test the device-independent performance of a system. Hence, it uses reference signatures obtained through one sensor, while the test samples (genuine and forgery) are obtained through another sensor. It is run only for the 20 people who are common in the two subcorpora.

The protocol proceeds as follows: the first 5 signatures of the Visual subcorpus is used as reference and the genuine signatures of the same users and all available forgeries in the Blind subcorpus are used as tests. Then, the first 5 signatures of the Blind subcorpus is used as reference and the genuine signatures of the same users in the Visual subcorpus are used as genuine tests. The EER should be calculated using a *single* threshold over the two experiments. Any tuning must be done before the experiments, but it may use either or both of the validation sets.

6.9 Performance Assessment

Systems using the SUSIG database and protocols should report the EER obtained using a particular protocol. They may also additionally provide ROC or error trade-off curves of their system, for these protocols. As there are many protocols associated with the SUSIG database, it would be desirable that the results of the base protocol(s) are included when reporting results on the SUSIG database.

Note that in addition to reporting the EER value computed exactly as specified by a particular protocol, it is possible to run cross validation tests in accordance with the protocol and report mean and standard deviation of the experiments. For instance, one can randomly choose any 5 signatures of one session as reference (not necessarily the first 5) and use the remaining genuine signatures of the same session

and all forgeries, as test signatures. Considering the Single Session protocol of the Visual subcorpus, one can run as many as $2(\frac{10!}{5!5!}) = 504$ experiments. However, given that there are several thousands signatures to test in each protocol, we believe that the suggested protocols are sufficient, as specified.

6.10 Benchmark Results

In order to comment on the relative difficulties of the various protocols and provide a benchmark, we report results of our signature verification system [98] on the SUSIG database. This system is the winner of the First International Signature Verification Competition, SVC2004 [104].

Table 6.10 lists the EER results of our benchmark system using the SUSIG protocols. In particular, the results for the base protocol of the Visual Subcorpus is 2.10% ERR. The error-tradeoff curve for this protocol is given in Figure 6.4.

Protocol	**Subcorpus**	**EER (%)**
Single-Session	Visual	1.41
Across-Session	Visual	2.12
Base	Visual	**2.10**
Skilled Forgery	Visual	0.30
Highly Skilled Forg.	Visual	3.36
Random Forgery	Visual	**4.08**
Base	Blind	**2.85**
Random Forgery	Blind	**2.82**

Table 6.4: Results of the base system for the SUSIG database and protocols. The protocols marked in **bold** are the essential protocols, while the others measure performance under certain restricted conditions.

As expected, the single session protocol had lower error rate compared to the across-session protocol (1.41% versus 2.12%) and the base protocol result was a mix of the single and across-session protocols, as expected. In fact, the base protocol

result is very close to the across-session protocol (2.10%), but this is not surprising because there are twice as many genuine signatures from the second session compared to the first.

Analysis of the results for skilled and highly skilled forgeries (0.3% and 3.36%) show that highly skilled forgery set is significantly more difficult, which is to be expected, given the amount of information provided to the forgers while collecting that set.

The relative difficulty of the Blind subcorpus, as indicated by the results of comparable protocols (1.41% vs. 2.85% and 0.3% vs. 2.85%) may be attributed to the lack of visual feedback in the collection of the Blind subcorpus.

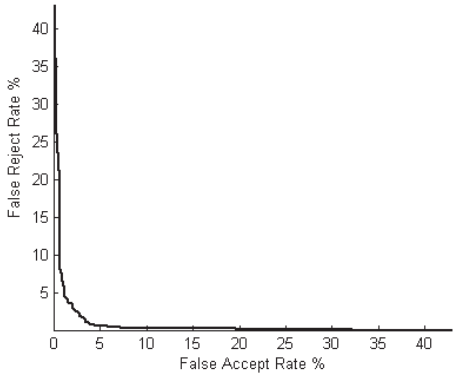

Figure 6.4: The error tradeoff curve indicates verification performance for different thresholds using the SUSIG Base Protocol of the Visual subcorpus.

On the other hand, the fact that random forgery results are higher than comparable skilled forgery tests is not very intuitive, since one would expect random forgery results to be much lower; after all, these are not even true forgeries but other people's genuine signatures. This is partly an artefact of our benchmark system which puts a significant emphasis on the correct timing of a signature: analysis of the random forgery errors has shown that intentional forgeries in the skilled and highly skilled sets are on average twice longer in duration compared to genuine signatures. Also, we have found that for some people with highly varying signatures, random

forgeries are as likely to be accepted as the highly skilled forgeries. In particular, 2 people out of 100 account for more than 10% of all the false accepts in the random forgery test.

Task	**Average EER (%)**
SVC2004 (Task1-Skilled)	2.84
SVC2004 (Task2-Skilled)	2.89
SVC2004 (Task1-Random)	2.79
SVC2004 (Task1-Random)	2.51

Table 6.5: Average EER obtained by our benchmark system in the SVC2004 competition.

In order to give some perspective on the results obtained with the SUSIG protocols, we also provide the results obtained by our benchmark system in the SVC2004 competition [104], summarized in Table 6.10. The SVC2004 database consists of made-up signatures which the contributors practiced solely for the sake of this database. In other aspects (size, number of sessions, tablet), the database is similar to the Blind Subcorpus, as explained in Section 6.2. The SVC2004 competition had two tasks: in Task1, competitors were given only the coordinate information for a signature, while in Task2 more features were supplied (coordinate, timestamp, azimuth, altitude, and pressure information). Our system was the winner of the skilled forgery tests of both tasks, with 2.84% and 2.89% average equal error rates over random trials using different genuine signatures as reference. The system ranked fourth and second in the random forgery tests of Task1 and Task2, with 2.79% and 2.51% average equal error rates, respectively. These results are quite similar to the results obtained in the the comparable protocols (base and random forgery protocols) of the Blind subcorpus.

Figures 6.5 and 6.6 show sample signatures that are very consistent and difficult to forge and signatures that are inconsistent and consequently easier to forge. The consistent group had zero false accept in random and skilled forgery tests listed in

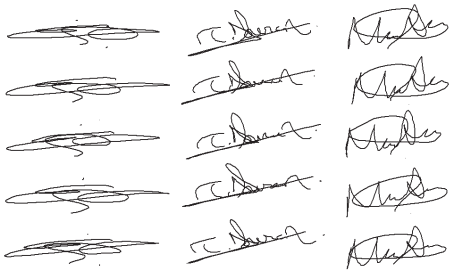

Figure 6.5: Sample genuine signatures of 3 subjects who are very consistent; these subjects were not forged at all in random or skilled forgery tests.

Table 6.10 while the inconsistent group had an average of 34.53% FAR (910, 663 and 478 false accepts out of 99x20=1980 random forgery attacks. This comparison highlights the fact that signature is a biometric with varying and adjustable complexity. Assuming that a person can be consistent in his/her signatures with some care, one could increase the complexity of his/her signature (duration, number of strokes etc.) and make it more difficult to forge, for higher security applications.

Figure 6.6: Sample genuine signatures of 3 subjects who are very inconsistent; these subjects had a high false accept rate. These signatures were forged 910, 663 and 478 times, from top to bottom, in 1980 random forgery attacks for each.

6.11 Summary

In this part of the thesis, an online signature database and associated protocols are presented. The database consists of two parts: the Visual subcorpus is collected using a pressure sensitive pad with a built-in LCD screen and consists of signatures provided by 100 subjects, for a total of 3000 signatures (2000 genuine and 1000 forgery). The Blind subcorpus is collected using a pressure sensitive tablet without an LCD screen. Each subject in this part provided 8 to 10 genuine signatures, for a total of 1940 signatures (940 genuine and 1000 forgery). Together with the validation data which is separate, the overall database contains slightly over 5000 signatures.

Signatures in the database are real signatures of the contributors, as opposed to made-up ones. Furthermore, signatures are collected over two different sessions, with at least one week delay between the sessions. Availability of real signatures and the fact that genuine signatures were not collected in a single session, make the database more realistic compared to some of the existing databases.

The method used to collect highly skilled forgeries is novel and found to result in challenging forgeries. Specifically, we developed a signature animation tool that animates the signature to be forged on the LCD screen of the tablet such that the forger can *trace* over the signature. We also provided a degree of similarity between forgeries and the corresponding reference sets so that forgers could improve over time. This highly skilled forgery set has significantly higher error rates compared to the skilled forgery set, thus providing researchers with more challenging forgeries. Note that since we cannot obtain real forgeries, the best we can do is to improve the forgery qualities.

Together with this database, we have established a set of protocols, paralleling other databases and common practice, so as to provide a benchmark for all needs (e.g. single-session protocol, multi-session protocol etc.). These clearly specified protocols will help in comparing performances of different systems.

Finally, to provide a benchmark, we measured the performance of our own system

[98] on the database and associated protocols. This system obtained 2.10% and 2.85% EER on the base protocols of the Visual and Blind Subcorpora, respectively. The same system obtained similar results with average equal error rates of 2.86% and 2.89% on the skilled forgery tests in SVC2004 [104].

One may argue that the lack of the azimuth and altitude features constitute a shortcoming for the SUSIG database. However, tablets with these features are relatively less common. Also, given the fact that the EER results obtained in Task2 of SVC2004 (where these features were provided), are similar to that of Task1 (coordinate information only), one may argue that they are not too important after all.

Chapter 7

Conclusions and Contributions

Biometric authentication systems are gaining popularity and being widely deployed. Such systems are being preferred over the traditional password and token-based authentication schemes, due to the security and convenience they provide. However there are increased concerns over the loss of privacy and potential misuse of biometric data. Major concerns are due to the facts that: i) strong biometrics (eg. iris, fingerprint) are highly unique to a person thus can be used for tracking individuals and their behavior, ii) biometric data kept in central databases may be compromised, which is a serious concern given the fact that if a biometric trait is stolen or compromised it can not be reissued or canceled, and iii) biometric data may disclose sensitive information such as race, gender and health problems. In this dissertation we addressed privacy concerns associated with biometrics systems and proposed practical solutions to alleviate some of the existing problems.

One of the main contributions of this thesis is the proposal of a novel privacy enhancing biometric authentication framework that is based on the idea of combining multiple biometrics to enhance both privacy and security. Specifically, two biometric traits (e.g. two fingerprints) are combined at the template level, in order to obtain a non-unique biometric identifier for a person. We demonstrated two different realizations of this framework; using two separate fingerprints in the first one and combining fingerprint and voice in the second one. We empirically showed that both realizations of the framework are more successful compared to their single biometric counterparts, while privacy is preserved to a large extent. The privacy is

measured as the ratio of the correct templates returned (precision) when searching the template database using only a single biometric trait. In other words, we have shown that using a single biometric trait, we cannot reliably locate the correct multi-biometric template created with this scheme. Furthermore, these multi-biometric identifiers are more successful in identity verification compared with single biometric counterparts. Finally, if compromised, multi-biometric identifiers can be revoked by providing different pairs of traits. This is especially true for the fingerprint and voice combination, since voice utterances can naturally incorporate a password. Likewise, different identifiers can be generated for different security applications, which will result in different identifiers. Since non-unique identifiers cannot be used for linking separate biometric databases, this eliminates concerns of tracking of personal activity through the use of biometrics. One another important property of our framework is that it can be used in applications requiring authorization of multiple users (eg. authorization of both husband and wife to withdraw money from shared bank account). In this case, the multi-biometric identifier is simply constructed from the biometrics of each party.

Another contribution of the thesis is the work done around the Fuzzy Vault scheme, which emerged as a privacy preserving framework that can be applied to biometrics. As applied to biometrics, it successfully combines biometrics and cryptography in order to get the benefits of both fields; while biometrics provides non-repudiation and convenience, cryptography guarantees privacy (based on the difficulty of the polynomial reconstruction problem) and adjustable levels of security. We have done a straightforward extension of the first implementation of the Fuzzy Vault using fingerprints, as well as providing the first realization using online signatures.

Apart form single person authentication, there are circumstances requiring presence of multiple users to initiate certain services or get access to a particular entity. In the cryptographic realm, such scenarios are called *secret sharing*. We demonstrated how the Fuzzy Vault scheme can be used for secret sharing, using biometric traits. Specifically, we showed that the key which is locked in a fuzzy vault using

the fingerprints of multiple people, can only be released when the required number of people present their individual biometrics.

Our last contribution in the area of Fuzzy Vault, is the empirical substantiation of a recent claim that the Fuzzy Vault scheme is susceptible to the correlation attacks, revealing the biometric traits if two different fuzzy vaults, created using same biometric trait but different chaff points, are intercepted. We implemented this attack, which involved non-trivial steps, and showed that the claim is indeed substantiated.

The discriminative capability of a biometric modality is based on its entropy and is an important factor in choosing a biometric for a large-scale deployment or a cryptographic application. Signature is a widely accepted and frequently adopted biometric trait. In order to further substantiate their applicability in biometric verification systems, we developed an individuality model for online signatures. We build our model on the probability of coincidence match between two arbitrary signatures. We proposed the Fourier domain representation of the signature along with a non-trivial matching algorithm to ease probability estimations. Using a large signature database, we estimated model parameters and demonstrated using this preliminary individuality model that an average online signature has a high level of complexity.

Throughout our signature related research we have collected a large online signature database. This database and associated protocols are made public to serve the research community. The protocols we have established are comprehensive and well-defined, and aim to be used in testing different biometric verification problems. For this database, we have devised a method to increase the quality of forgeries, so as to have more challenging forgeries. Finally, we have published the results of our online signature verification algorithm on this database, to provide a benchmark. The collection of such databases is time and budget consuming, but they are crucial in order to objectively assess performance results of different methods; as such, this constitutes a smaller contribution of this thesis.

Bibliography

[1] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication," in *Proceedings of the IEEE*, **91**(12), pp. 2021–2040, Dec. 2003.

[2] "Passwords revealed by sweet deal," 20 April, 2004. BBC News, http://news.bbc.co.uk/1/hi/technology/3639679.stm.

[3] "2004 annual password survey results," 2004. SafeNet Inc., http://www.safenet-inc.com.

[4] J. Woodward, "Biometrics: Privacy's foe or privacy's friend?," *Proceedings of the IEEE* **85(9)**, p. 1487, 1997.

[5] W. H. I. McLean, "Genetic disorders of palm skin and nail," *Journal of Anatomy* **202(1)**, pp. 133–141, 2003.

[6] H. Chen, *Medical Genetics Handbook*, St. Louis, MO: W.H. Green, 1988.

[7] M. M. Schuster, "Gastroenterology: Fingerprinting gi disease," in *Johns Hopkins Physician Update*, p. 5, Apr. 1996.

[8] B. Bates, "A guide to physical examination and history taking," in *5th ed. Philadelphia, PA: Lippincott*, pp. 181–215, 1991.

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

[10] U. Uludag, S. Pankanti, S. Prabhakar, and A. Jain, "Biometric cryptosystems: Issues and challenges," in *Proceedings of the IEEE*, **92**(6), 2004.

[11] C. Soutar, D. Roberge, S. Stojanov, R. Gilroy, and B. V. Kumar, "Biometric encryption using image processing," *In Proc. SPIE, Optical Security and Counterfeit Deterrence Techniques II* **Vol. 3314**, pp. 178–188, 1998.

[12] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data," in *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, 2004.

[13] A. Juels and M. Sudan, "A fuzzy vault scheme," in *Proceedings of IEEE International Symposium on Information Theory*, p. 408, 2002.

[14] A. Teoh, D. Ngo, and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognition* **37**, pp. 2245–2255, 2004.

[15] A. Teoh and D. Ngo, "Cancellable biometrics featuring with tokenized random number," *Pattern Recognition Letters* **26**(10), pp. 1454–1460, 2004.

[16] A. Teoh and D. Ngo, "Integrated wavelet and fourier-mellin invariant feature in fingerprint verification system," in *Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications*, pp. 82–88, 2003.

[17] G. Davida, Y. Frankel, and B. Matt, "On enabling secure applications through on-line biometric identification," in *IEEE Symposium on Privacy and Security*, p. 148157, 1998.

[18] G. Davida, Y. Frankel, B. Matt, and R. Peralta, "On the relation of error correction and cryptography to an off-line biometric based identification scheme," in *Proceedings of International Workshop Coding and Cryptography (WCC99)*, 1999.

[19] F. Hao, R. Anderson, and J. Daugman, "Combining crypto with biometrics effectively," *IEEE Transactions on Computers* **55**(9), 2006.

[20] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(11), pp. 1148–1161, 1993.

[21] S. Agaian, *Hadamard Matrix and Their Applications*, Springer Verlag, 1985.

[22] N. MacWilliams, F.J.and Sloane, *The Theory of Error-Correcting Codes*, North Holland, 1991.

[23] F. Monrose, M. Reiter, and S. Wetzel, "Password hardening based on keystroke dynamics," *International Journal of Information Security* , 2001.

[24] F. Monrose, M. Reiter, Q. Li, and S. Wetzel, "Cryptographic key generation from voice," in *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, May 2001.

[25] F. Monrose, M. Reiter, Q. Li, and S. Wetzel, "Using voice to generate cryptographic keys," in *The Speech Recognition Workshop*, June 2001.

[26] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Conference on Computer and Communications Security, ACM Press.*, pp. 28–36, 1999.

[27] C. Clancy, N. Kiyavash, and D. Lin, "Secure smartcard - based fingerprint authentication," in *ACM Workshop on biometric methods and applications*, Nov. 2003.

[28] S. Yang and I. Verbauwhede, "Secure fuzzy vault based fingerprint verification system," *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on* **1**, pp. 577–581, 2004.

[29] U. Uludag, S. Pankanti, and A. Jain., "Fuzzy vault for fingerprints," in *Proceeding of International Conference on Audio and Video Based Biometric Per-*

son Authentication, pp. 310–319, 2005.

[30] H. Feng and C. Wah, "Private key generation from on-line handwritten signatures," *Information Management and Computer Security,* **10/4**, pp. 159–164, 2002.

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

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

[33] E. Verbitskiy, P. Tuyls, D. Denteneer, and J. Linnartz, "Reliable biometric authentication with privacy protection," in *Proceedings of Benelux Symposium on Information Theory*, May 2003.

[34] J. Linnartz and P. Tuyls, "New shielding functions to enhance privacy and prevent misuse of biometric templates," in *Proceeding of International Conference on Audio and Video Based Biometric Person Authentication*, **LNCS 2688**, pp. 393–402, 2003.

[35] P. Tuyls, E. Verbitskiy, T. Ignatenko, D. Denteneer, and T. Akkermans, "Privacy protected biometric templates: Acoustic ear identification," in *Proceedings of SPIE: Biometric Technology for Human Identification*, **Vol. 5404**, pp. 176–182, 2004.

[36] M. Henrik, H. Dorte, J. Boje, and S. Friis, "Transfer characteristics of headphones measured on human ears," *Journal of Audio Engineering Society* **43**(4), pp. 203–217, Apr. 1995.

[37] R. Tyler, "Britain is video surveillance capital of the world," 06 Dec. 2006. http://www.wsws.org/.

[38] C. Ramsey, "Limited authorization of video surveillance and privacy protection act of 2002 (bill 14-946)." Chief of Police, Metropolitan Police Department.

[39] A. Lecours, "Surveillance cameras in work environment and privacy," in *Labor Law in Canada Newsletter*, Canada, 2006.

[40] "Electronic privacy information center: Video surveillance." http://epic.org/privacy/surveillance.

[41] E. Newton, L. Sweeney, and B. Malin, "Preserving privacy by de-identifying facial images," tech. rep., CMU-CS-03-119, 2003.

[42] "Video image display apparatus and method." US Patent granted to Sony Inc., http://www.patentstorm.us/patents/6476820-claims.html.

[43] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Y. Tian, and A. Ekin, "Blinkering surveillance: Enabling video privacy through computer vision," tech. rep., IBM, Vol: RC22886, 2003.

[44] T. Boult, "Pico: Privacy through invertible cryptographic obscuration," in *IEEE/NFS Workshop on Computer Vision for Interactive and Intelligent Environments*, 2005.

[45] F. Dufaux and T. Ebrahimi, "Scrambling for video surveillance with privacy," in *Proceedings of IEEE Workshop on Privacy Research In Vision*, 2006.

[46] F. Dufaux, M. Ouaret, Y. Abdeljaoued, A. Navarro, F. Vergnenegre, and T. Ebrahimi, "Privacy enabling technology for video surveillance," in *Proceedings of SPIE*, **6250**, 2006.

[47] W. Zhang, S. Cheung, and M. Chen, "Hiding privacy information in video surveillance system," in *IEEE International Conference on Image Processing*, 2005.

[48] P. Phillips, "Privacy operating characteristic for privacy protection in surveillance applications," in *Audio- and Video-Based Biometric Person Authentication*, **3546**, 2005.

[49] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer, New York, 2003.

[50] M. Kucken, "Models for fingerprint pattern formation," *Forensic Science International, Elsevier* **171**(2-3), pp. 85–96, Sep. 2007.

[51] A. Jain., L. Hong, S. Pankanti, and R. Bolle, "An identity authentication system using fingerprints," *Proceedings of the IEEE* **85**, pp. 1365–1388, Sep. 1997.

[52] N. Ratha, S. Chen, and A. Jain., "Adaptive flow orientation based feature extraction in fingerprint images," *Pattern Recognition* **28**, pp. 1657–1672, 1995.

[53] S. Prabhakar, Wang, A. J., Jain, S. Pankanti, and R. Bolle, "Minutiae verification and classification for fingerprint matching," in *Proceedings of 15'th International Conference on Pattern Recognition*, **1**, pp. 25–29, Sep. 2000.

[54] M. Tico and P. Kuosmanen, "An algorithm for fingerprint image postprocessing," in *Proceedings of the 34'th Asilomar Conference on Signals, Systems and Computers*, **2**, pp. 1735–1739, Nov. 2000.

[55] M. Pradhan, "Comparative approach for minimum number of points of agreement required establishing identical prints," *SCIENTIFIC WORLD, A Multidisciplinary Annual Journal of Science and Technology* **4**(4), pp. 113–115, July 2006.

[56] A. Jain, L. Hong, and R. Bolle, "On-line fingerprint verification," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(4), pp. 302–314, 1997.

[57] N. Ratha and V. Pandit, "Robust fingerprint authentication using local struc-

tural similarity," in *Proceedings of the 5'th IEEE workshop on applications of computer vision*, pp. 29–34, Dec. 2000.

[58] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain, "First international competition for fingerprint verification algorithms," in *15'th International Conference on Pattern Recognition*, 2000. http://bias.csr.unibo.it/fvc2000/.

[59] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain, "Second international competition for fingerprint verification algorithms," in *16'th International Conference on Pattern Recognition*, 2002. http://bias.csr.unibo.it/fvc2002/.

[60] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain, "Third international competition for fingerprint verification algorithms," in *First International Conference on Biometric Authentication*, 2004. http://bias.csr.unibo.it/fvc2004/.

[61] D. Maio, D. Maltoni, R. Cappelli, J. Wayman, and A. Jain, "Fourth international competition for fingerprint verification algorithms," in *BioSecure Network of Excellence; FP6 IST-2002-507634*, 2006. http://bias.csr.unibo.it/fvc2006.

[62] C. Wilson, R. Hicklin, M. Bone, H. Korves, P. Grother, B. Ulery, R. Micheals, M. Zoepfl, S. Otto, and C. Watson, "Fingerprint vendor technology evaluation 2003: Summary of results and analysis report," in *National Institute of Standards and Technology*, **NISTIR 7123**, June 2004. http://fpvte.nist.gov/.

[63] A. Ross and A. Jain, "Multimodal biometrics: an overview," in *Proceeding of European Signal Processing Conference*, pp. 1221–1224, Sep. 2004.

[64] R. Snelick, U. Uludag, A. Mink, M. Indovina, and A. Jain, "Large-scale evaluation of multimodal biometric authentication using state-of-the-art systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*

27(3), March 2005.

[65] R. Brunelli and D. Falavigna, "Person identification using multiple cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(10), pp. 955–966, Oct. 1995.

[66] S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz, "Fusion of face and speech data for person identity verification," in *IEEE Transactions on Neural Networks*, **10**(5), pp. 1065–1075, 1999.

[67] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(3), pp. 226–239, Mar. 1998.

[68] L. Hong and A. Jain, "Integrating faces and fingerprints for personal identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(2), Dec. 1998.

[69] E. Camlikaya, A. Kholmatov, and B. Yanikoglu, "Multimodal biometric templates for verification using fingerprint and voice," in *SPIE Defense and Security: Biometric Technology For Human Identification V*, March 2008.

[70] S. Young, J. Jansen, J. Odell, D. Ollason, and P. Woodland, *The HTK Book for HTK Version 2.1*, Cambridge University Press, 1997.

[71] E. Camlikaya, B. Yanikoglu, and H. Erdogan, "Voice verification using hmm," in *IS&T/SPIE Annual Symposium on Electronic Imaging*, Jan. 2008.

[72] "Advanced encryption standard (AES)," *National Institute of Standards and Technology)* , Nov. 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[73] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM* **21**(2), pp. 120–126, 1978.

[74] J.-J. Brault and R. Plamondon, "A complexity measure of handwritten curves: modeling of dynamic signature forgery," *Systems, Man and Cybernetics, IEEE Transactions on* **23(2)**, pp. 400–413, 1993.

[75] A. Shamir, "How to share a secret," *Communications of the ACM* **22**, pp. 612–613, 1979.

[76] W. J. Scheirer and T. E. Boult, "Cracking fuzzy vaults and biometric encryption," in *Univ. of Colorado at Colorado Springs, Technical Report*, February 2007.

[77] A. Kholmatov, B. A. Yanikoglu, E. Savas, and A. Levi, "Secret sharing using biometric traits," in *Biometric Technology For Human Identification III, Proceedings of SPIE*, **6202**, 18 April, 2006.

[78] K. Nandakumar, A. Nagar, and A. Jain, "Hardening fingerprint fuzzy vault using password," in *International Conference on Biometrics*, pp. 927–937, 2007.

[79] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal* **27**, pp. 379–423, 623–656, July, October 1948.

[80] J. Massey, "Guessing and entropy," in *IEEE International Symposium on Information Theory, Trondheim, Norway*, p. 204, 1994.

[81] C. Cachin, *Entropy measures and unconditional security in cryptography*. PhD thesis, Swiss Federal Institute of Technology, Zurih, 1997.

[82] S. Srihari, S. Cha, and S. Lee, "Establishing handwriting individuality using pattern recognition techniques," in *ICDAR '01: Proceedings of the Sixth International Conference on Document Analysis and Recognition*, p. 1195, 2001.

[83] S. Srihari, S. Cha, H. Arora, and S. Lee, "Individuality of handwriting: A validation study," in *ICDAR '01: Proceedings of the Sixth International Conference on Document Analysis and Recognition*, p. 106, 2001.

[84] H. Kuwabara and T. Takagi, "Acoustic parameters of voice individuality and voice-quality control by analysis-synthesis method," *Speech Communication-Volume* **10**(5-6), pp. 491–495, 1991.

[85] H. Kuwabara, "A perceptual experiment on voice individuality by altering pitch and formant frequencies," *The Journal of the Acoustical Society of America* **100**(4), pp. 2600–2600, 1996.

[86] C. Vielhauer, R. Steinmetz, and A. Mayerhofer, "Biometric hash based on statistical features of online signatures," in *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 1*, p. 10123, 2002.

[87] C. Vielhauer and R. Steinmetz, "Handwriting: Feature correlation analysis for biometric hashes," *EURASIP Journal on Applied Signal Processing* **2004**(4), pp. 542–558, 2004.

[88] S. Pankanti, S. Prabhakar, and A. Jain, "On the individuality of fingerprints," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(8), pp. 1010–1025, 2002.

[89] J. Chen and Y. Moon, "A minutiae-based fingerprint individuality model," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–7, 2007.

[90] R. Bolle, S. Pankanti, J. Connell, and N. Ratha, "Iris individuality: A partial iris model," in *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2*, pp. 927–930, 2004.

[91] S. Yoon, S. Choi, C. S., L. Y., and C. Tappert, "On the individuality of the iris biometric," *ICGST International Journal on Graphics, Vision and Image Processing* **Vol.5**, pp. 63–70, 2005.

[92] L. Ballard, F. Monrose, and D. Lopresti, "Biometric authentication revisited: understanding the impact of wolves in sheep's clothing," in *USENIX-SS'06:*

Proceedings of the 15th conference on USENIX Security Symposium, pp. 3–3, 2006.

[93] L. Ballard, D. Lopresti, and F. Monrose, "Evaluating the security of handwriting biometrics," in *In proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR06)*, 2006.

[94] A. Jain, F. Griess, and S. Connell, "On-line signature verification," *Pattern Recognition* **35**, pp. 2963–2972, December 2002.

[95] C. Vielhauer, R. Steinmetz, and A. Mayerhofer, "Biometric hash based on statistical features of on-line signatures," in *Proceedings of the 16'th International Conference on Pattern Recognition*, **1**, p. 10123, 2002.

[96] T. Ohishi, Y. Komiya, and T. Matsumoto, "On-line signature verification using pen-position, pen-pressure and pen-inclination trajectories," in *Proceedings of the 15'th International Conference on Pattern Recognition*, **4**, p. 4547, 2000.

[97] X. Yang, T. Furuhashi, K. Obata, and Y. Uchikawa, "Constructing a high performance signature verification system using a ga method," in *2nd New Zealand Two-Stream International Conference on Artificial Neural Networks and Expert Systems (ANNES '95)*, pp. 170–173, 1995.

[98] A. Kholmatov and B. Yanikoglu, "Identity authentication using improved online signature verification method," *Pattern Recognition Letters* **26**(15), pp. 2400–2408, 2005.

[99] R. Martens and L. Claesen, "Dynamic programming optimisation for on-line signature verification," in *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, **2**, pp. 653–656, 1997.

[100] M. Parizeau and R. Plamondon, "A comparative analysis of regional correlation, dynamic time warping and skeletal tree matching for signatures," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence **12**(7), pp. 710–717, 1990.

[101] J. J. Van Oosterhout, H. Dolfing, and E. Aarts, "On-line signature verification with hidden markov models," in *Proceedings of the 14'th International Conference on Pattern Recognition*, **2**, p. 1309, 1998.

[102] R. Plamondon and G. Lorette, "Automatic signature verification and writer identification - state of the art," *Pattern Recognition* **22**(2), pp. 107–131, 1989.

[103] F. Leclerc and R. Plamondon, "Automatic signature verification: The state of the art," *International Journal of Pattern Recognition and Artificial Intelligence* **8**(3), pp. 643–660, 1994.

[104] D. Yeung, H. Chang, Y. Xiong, S. George, R. Kashi, T. Matsumoto, and G. Rigoll, "SVC2004: First international signature verification competition," in *Proceedings of the International Conference on Biometric Authentication*, **1**, pp. 16–22, 2004. Also available at `http://www4.comp.polyu.edu.hk/~icba/`.

[105] S. J. Elliott and A. R. Hunt, "Proceedings of the 6th the challenge of forgeries and perception of dynamic signature verification," in *International Conference on Recent Advances in Soft Computing (RASC 2006), K. Sirlantzis (Ed.)*, pp. 455–459, 2006.

[106] A. Kholmatov and B. Yanikoglu, "Fourier descriptors for on-line signature verification," *under review* , 2007.

[107] R. Gonzalez and R. Woods, *Digital Image Processing*, pp. 497–502. Addison-Wesley, 1992.

[108] A. Kholmatov and B. Yanikoglu, "Susig database," 2007. http://biometrics.sabanciuniv.edu.

[109] J. Ortega-Garcia, "Mcyt baseline corpus: A bimodal biometric database," in *IEE Proceedings Vision, Image and Signal Processing, Special Issue on*

Biometrics on the Internet, **150(6)**, pp. 395–401, 2003.

[110] S. Garcia-Salicetti, C. Beumier, G. Chollet, B. Dorizzi, J. Leroux-Les Jardins, J. Lunter, Y. Ni, and D. Petrovska-Delacretaz, "Biomet: a multimodal person authentication database including face, voice, fingerprint, hand and signature modalities," in *Proc. of 4'th International Conference on Audio and Video-Based Biometric Person Authentication*, pp. 845–853, 2003.

[111] B. Li, D. Zhang, and W. K., "Online signature verification based on null component analysis and principal component analysis," *Pattern Analysis and Applications* **8**, pp. 345–356, 2006.

[112] J. Richiardi, H. Ketabdar, and A. Drygajlo, "Local and global feature selection for on-line signature verification," in *Proceedings of the 12th International Conference on Document Analysis and Recognition*, pp. 625–629, 2005.