SECURE BIOMETRIC AUTHENTICATION WITH FIXED-LENGTH BINARY REPRESENTATIONS

Thesis submitted in partial fulfillment
of the requirements for the degree of

MS by Research
in
Computer Science

by

Rohan Kulkarni
200702017
rohan.kulkarni@research.iiit.ac.in

CVIT
International Institute of Information Technology
Hyderabad - 500 032, INDIA
December 2014

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Secure Biometric Authentication with Fixed-Length Binary Representations" by Rohan Kulkarni, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date

Adviser: Prof. Anoop Namboodiri

To My Family

Acknowledgments

I would sincerely like to thank my advisor Prof. Anoop Namboodiri for his guidance and support. It steered the direction of my work and motivated me to work hard. The freedom to choose topics to work on, the discussions that followed and his working style helped me to carry out my research without any pressures, also gain a sense of responsibility. I would also like to thank Prof. Srinathan Kannan for his guidance and support during my early stages of work. The discussions with him on various topics, though majority of them not related to academics, helped me understand and observe things with a better perspective.

I would like to thank my friends Akhil, Kogta, Mayank, Parikshit, Prabhu, Pulkit, Sahni, and Siddhartha for the memorable time spent during my stay at IIIT-H. Last but not the least, I would like to thank my parents and my sister for their complete support and motivation to work hard.

Abstract

Biometrics have been established to be extremely reliable at the task of identifying individuals and thus are at the core of several real-world systems ranging from employee attendance to access control systems in the military. With growing computing resources available to individuals, biometric authentication systems are being deployed in even wider range of commercial applications. The permanent nature of biometrics raises serious security concerns with these deployments. Also, losing one's biometric trait can compromise that individual's identity in all the systems he is enrolled in. Biometrics are of non-rigid nature, requiring a fuzzy matching process, thus making it difficult to directly borrow popular security techniques used elsewhere with passwords and key-cards. Thus, the research interest received by this field attempts to develop efficient and reliable biometric authentication systems while addressing the issues of security and privacy.

Binary biometric representations have been shown to provide significant improvement in efficiency without compromising the system performance for various modalities including fingerprints, palmprints and iris. Hence, this thesis is focused on developing secure and privacy preserving protocols for fixed-length binary biometric templates which use hamming distance as the dissimilarity measure. We propose a novel authentication protocol using a *somewhat* homomorphic encryption scheme that provides template protection and ability to use masks while computing the hamming distance. The protocol operates on encrypted data, providing complete biometric privacy to individuals trying to authenticate, only revealing the final matching score to the server. It allows real-time authentication and retains matching accuracy of the underlying representation as demonstrated by our experiments on iris and palmprints.

We also propose a one-time biometric token based authentication protocol for widely used banking transactions. In the current scenario, the user is forced to trust the service provider with his sole banking credentials or credit card details for availing desired services. Often used one-time password based systems do provide additional transaction security, however the organizations using such systems are still incapable of differentiating between a genuine user trying to authenticate or an adversary with stolen credentials. Involving biometric security would certainly strengthen the authentication process. The proposed protocol upholds the requirements of secure authentication, template protection and revocability while providing user anonymity from the service provider. We demonstrate our system's security and performance using iris biometrics to authenticate individuals.

Contents

Chapter Page

1 Introduction . . . 1
1.1 Biometric system . . . 2
1.1.1 Modes of operation . . . 3
1.1.2 System vulnerabilities . . . 4
1.1.3 Effects of template compromise . . . 7
1.2 Motivation and contribution . . . 7
1.3 Organization of the thesis . . . 8

2 Background . . . 9
2.1 Template protection schemes . . . 10
2.1.1 Cancelable biometric transformations . . . 10
2.1.1.1 Non-invertible transforms . . . 11
2.1.1.2 Biometric salting . . . 12
2.1.2 Biometric cryptosystems . . . 12
2.1.2.1 Key-generating systems . . . 14
2.1.2.2 Key-binding systems . . . 14
2.2 Cryptographic protocols . . . 16
2.2.1 Two-party computation . . . 17
2.2.2 Homomorphic encryption . . . 17
2.3 Summary . . . 18

3 Secure Hamming Distance Based Biometric Authentication . . . 19
3.1 Introduction and related work . . . 19
3.2 Preliminaries . . . 21
3.3 Homomorphic encryption . . . 21
3.3.1 BGN cryptosystem . . . 22
3.4 Iris matching . . . 23
3.5 Proposed protocol . . . 24
3.6 Protocol analysis . . . 28
3.6.1 Correctness . . . 28
3.6.2 Computational analysis . . . 28
3.6.3 Security and privacy analysis . . . 29
3.7 Experiments . . . 31
3.7.1 Implementation and results . . . 31
3.7.2 Alternate method . . . 31

3.8 Summary 32

4 One-Time Biometric Token based Authentication 33
4.1 Introduction and related work 33
4.2 One-time passwords 34
4.3 Error correction 35
4.4 Authentication protocol 36
4.4.1 Registration phase 37
4.4.2 Authentication phase 37
4.5 Experiments 39
4.6 Protocol analysis 40
4.6.1 Correctness 40
4.6.2 Privacy analysis 41
4.6.3 Attacks and security analysis 41
4.7 Summary 42

5 Conclusion and future work 44

Bibliography 46

List of Figures

Figure		Page
1.1	Common biometric traits used for identifying individuals.	2
1.2	General structure of a biometric system. (a) Enrollment mode: An individual's biometric data is stored in the system. (b) Authentication mode: An individual is verified based on a currently captured biometric sample with the earlier stored template.	3
1.3	Attack model for a general biometric system.	6
2.1	Some Non-Invertible Transforms applied to biometric features.	10
2.2	A Biohashing Scheme based on convolving the feature matrix onto a orthogonal matrix.	12
2.3	A Fuzzy Commitment Encryption Protocol.	13
2.4	Fuzzy Commitment Decryption Protocol.	13
2.5	Fuzzy Vault Encryption Protocol.	15
2.6	Fuzzy Vault Decryption Protocol.	16
3.1	Protocol for secure hamming distance based biometric authentication using masks.	25
4.1	General One-Time Password Authentication	34
4.2	One-Time Biometric Token	36
4.3	Proposed one-time biometric token based authentication protocol.	37

List of Tables

Table		Page
2.1	Comparison of various template protections schemes, their advantages and limitations.	17
3.1	Security, computation and bandwidth requirement with different pairing schemes.	31
3.2	Comparison with existing similar systems.	32
4.1	Matching performance of the proposed protocol on CASIA V3 Iris evaluation subset at different key lengths. (b - RS code block size)	40
4.2	Computation costs of the proposed protocol.	40

Chapter 1

Introduction

The term 'biometrics' is derived from the Greek words 'bios' which means *life* and 'metrics' which means *to measure*. The science of identifying individuals based on their anatomical and behavioral traits is known as biometric recognition. Humans have been identifying other humans as familiar and unfamiliar individuals using their face and voice since the beginning of civilization. This simple task gradually became complex as the civilization grew and humans traveled throughout the world. As the society became complex, several institutions evolved which required this task to be solved efficiently and accurately. Beginning with justice systems, which wanted to identify repeat offenders for a severe punishment than first time offenders, biometric identification techniques were also developed for paying employee wages, recording attendance etc. As technology advanced various computer based applications evolved to make life easier. Several of these applications provide individual specific utilities such as sending e-mails, storing and accessing digital documents, performing bank transactions and also accessing physically secured spaces based on appropriate authorization. Most of these are based on the use of key-cards or passwords to identify individuals. Use of biometrics in such systems certainly adds to the security as they address the problem of identifying *who* the individual '*is*' as opposed to a password based system which addresses *what* the individual '*knows*' and the token based system which addresses *what* the individual '*has*'.

The initial systems developed to recognize individuals, Bertillonage [1], used multiple body measurements i.e. the length and width of head, lengths of hands and fingers to record individuals and Galton [35] and Faulds [31] recorded the ridge patters in fingerprints. These initial developments were completely manual. Automated systems to acquire, store and match fingerprints [11, 33] were developed in 1960's. Systems based on other biometric traits were soon developed, face [10], palmprints [34] and iris [25]. With the advancement of technology efficient scanners and sensors were developed allowing high quality acquisition of biometric traits, also leading to development of efficient and accurate algorithms for identifying individuals. This has led to widespread use of biometrics from secure personal safes, hospitals to law enforcement agencies and the military which require high level of security. Figure 1.1 shows the common biometric traits used for identifying individuals. An example of a state of the art system is the iris based recognition system used by the UAE foreign ministry [4] at air,

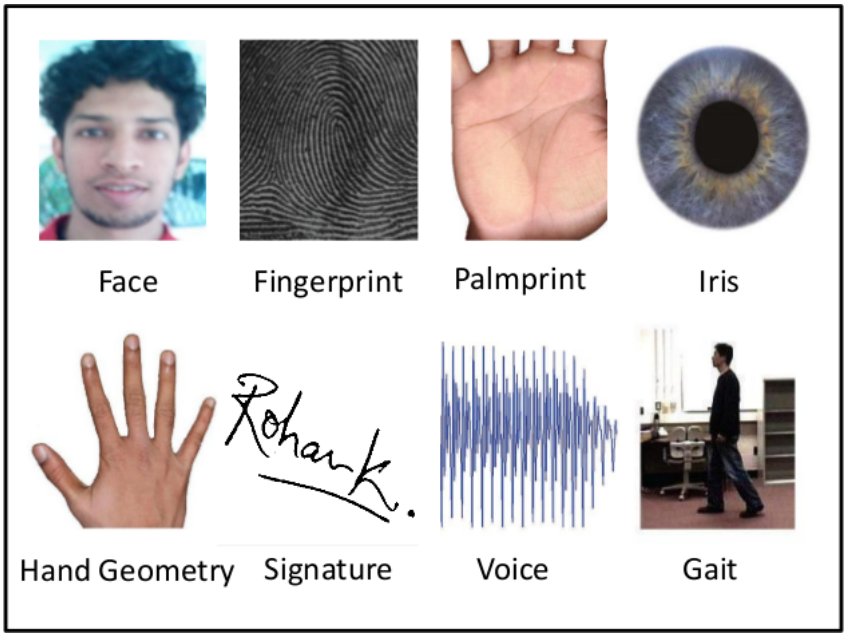

Figure 1.1 Common biometric traits used for identifying individuals.

land and sea ports which performs about 12 billion comparisons per day to identify individuals on their watch-list.

In the following sections, we will first define a biometric recognition system and describe its various modules. We then point out and analyze in detail the vulnerabilities of the biometric system. Next, we discuss the motivation and contribution of our work.

1.1 Biometric system

A pattern recognition system that recognizes individuals based on any of their biometric traits is a biometric recognition system, or simply a biometric system. This pattern recognition system must be capable of capturing an individual's biometric trait, store it in a way that it can classify different samples and perform a correct classification when used for recognizing that individual. Its structure and working has been discussed in detail by Jain *et al.* [42]. Generally it consists of the following modules: *(i)* Sensor, *(ii)* Feature extractor, *(iii)* Template database, *(iv)* Matcher and *(v)* Decision module. The sensor captures an instance of the biometric trait presented. There may be a separate module to access the quality of the acquired biometric instance and it may request for a re-capture if the quality is poor. The feature extractor extracts prominent features from the captured instance. During the enrollment phase, these features are stored in the template database. During the authentication phase, the matcher module outputs a score for the performed match between the features presented and the features stored. Based on a threshold value, the decision module outputs the result. The structure is illustrated in Figure 1.2

We discuss the terminology related to a biometric system below:

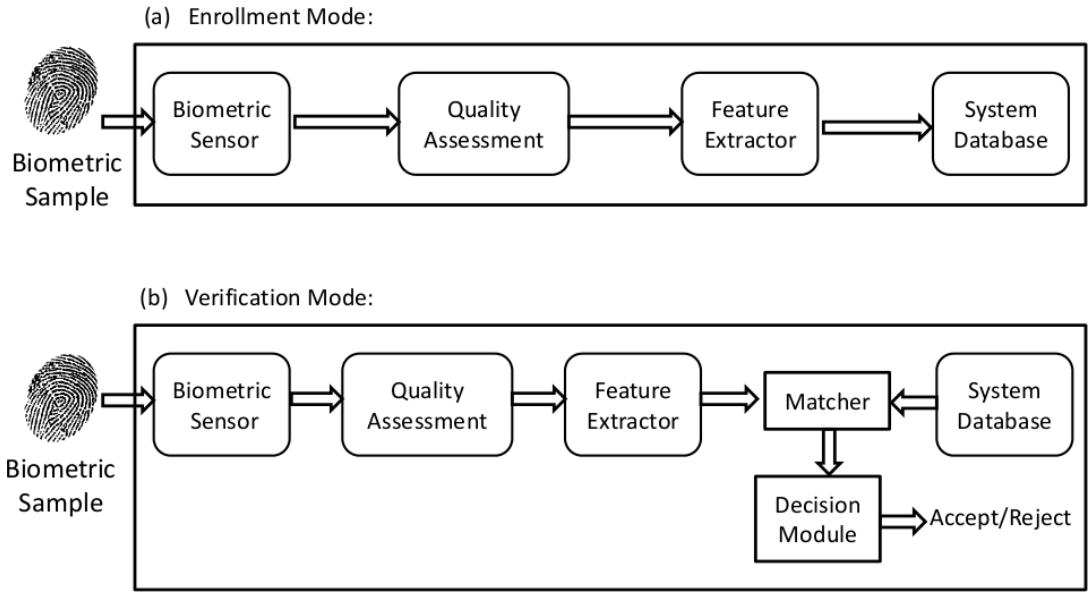

Figure 1.2 General structure of a biometric system. (a) Enrollment mode: An individual's biometric data is stored in the system. (b) Authentication mode: An individual is verified based on a currently captured biometric sample with the earlier stored template.

- The anatomical or behavioral trait of an individual used for his identification is called a **Biometric trait**. Fingerprints, iris, palmprints, face, gait, signature and voice are commonly used biometric traits (Figure 1.1).
- An instance of the specific biometric trait is called a **Biometric instance**. For example the right thumb, or the right eye.
- A captured sample of an instance of a biometric trait of an individual is called a **Biometric sample**.
- The data, stored in a system as features, extracted from a biometric sample of an individual during his enrollment is called a **Biometric template** or **Template**.
- The data extracted as features from a biometric sample of an individual during authentication which is compared with a template is called a **Biometric query** or **Query**.
- The minimum amount of similarity required between a query and a template to declare a match between them is called a **System threshold**.

1.1.1 Modes of operation

A typical biometric recognition system operates in two modes - enrollment and authentication. In the initial mode, the system captures a sample of a biometric trait of an individual enrolling with it. This

biometric sample is processed to extract a feature vector which is stored in the system database as a biometric template along with the identity of the individual. The authentication mode operations vary based on whether the system performs a verification or an identification. In a verification system the user provides his identity details along with the biometric trait's sample while authenticating himself. The features extracted from that query sample are matched only with the template stored against the claimed identity of the user. A matching score greater than the system threshold is considered as an identity match, otherwise a non-match is declared. Widely spread commercial biometric systems are mostly verification systems where each user is provided with an identity which he provides along with the biometric at the authentication phase. On the other hand, in an identification system the enrolled user just provides his biometric sample during authentication. The produced biometric query is matched with all the templates in the system database to generate matching scores. Usually if one of the templates matches with a score greater than the system threshold, it is declared as the match. Some identification systems are designed to generate a list of candidate matches for the query. Such systems output all the templates matched with a score greater than the threshold, then usually a manual match is performed on the candidates for declaring the identity. Government organizations usually require applications where the aim is to identify an individual from their database based solely on his captured biometric traits. Government of India's UID [3] system is a good example of such an application.

1.1.2 System vulnerabilities

Biometric systems too are vulnerable to internal failures or adversary attacks which can lead to intrusion by illegitimate users causing theft of sensitive user information on the system or theft of the enrolled biometric templates. Figure 1.3 provides an overview of the vulnerability points of a biometric system. We categorize and discuss them below.

- **System classifier failure**

  It is the genuine classification failure of the system during the authentication process. This misclassification performed is due to the poor performance rate of the system. Performance of the classifier is quantified by measuring its outputs on genuine and impostor matching. On each input, the system computes a similarity score which is compared with the threshold to output a binary decision. A higher similarity score than the threshold is considered as a match and a lower score is a non-match. The score obtained on providing genuine inputs i.e. both samples belonging to the same individual is a genuine score while the one otherwise is an impostor score. A reasonable separation between genuine scores and impostor scores allows the system to maintain high performance. The following metrics also help in deciding the performance of the system.

  - *False Match Rate(FMR):* The rate at which the system incorrectly matches two different individual's samples. It is also called the *False Accept Rate(FAR)* when a single matching attempt per authentication is permitted.

- *False Non-match Rate(FNMR):* The rate at which the system incorrectly declares a non-match between two samples of the same individual. It is also called the *False Reject Rate(FRR)* when a single matching attempt per authentication is permitted.

If false accept and false reject probabilities are high for a system, it is under serious threat. Simplest of the attacks i.e. providing any random instance of the desired biometric trait can get the adversary authenticated. This is also termed the *Zero Effort Attack*. On the other hand, if it is difficult to extract required features from the captured trait due to quality issues, a legitimate user may be denied access to the system.

Improving feature extraction and matching techniques, combining different biometric traits, using additional factors like smart-cards, passwords or security PINs can reduce the susceptibility internal failures. Tuning the threshold value to control genuine and impostor accept rates also helps in securing the internal failure vulnerability.

- **Insider attacks**

  The biometric system administrator, also called an insider, usually has privileges to handle the authentication process. In case of exceptions such as inability to acquire biometrics or during an alternate verification method in process, the adversary can collude with the administrator to let himself enroll or authenticate or force him to keep the system open for an attack.

  To limit such vulnerabilities regular auditing of the system and the administrators must be performed. Keeping the administrators anonymous also helps.

- **Infrastructure vulnerabilities**

  The hardware or software infrastructure of the biometric system can be targeted by an adversary. The attacks on infrastructure can be categorized as follows:

  - **Attacks on the user interface**

    The user interface is responsible for capturing the biometric trait using a sensor and transferring the information captured to the feature extraction module. An adversary can impersonate a genuine user by providing a spoof biometric of that individual, acquired through some covert means. On the other hand, an adversary avoiding to be identified through his biometrics can provide an improper sample at the interface. The biometric sensor at the interface can be tampered to perform attacks using the previously captured data or can be damaged to affect the quality of sample captured.

    To limit such attacks the sensor must be made sturdy to physical tampering and damage. Liveness detection techniques should be implemented to ensure presence of a live biometric trait. Thermal scanners to detect the temperature of fingerprints, infrared scanners to verify the warm blood in veins in the eye can help securing such attacks. Another alternative is to have physical security present at the user interface.

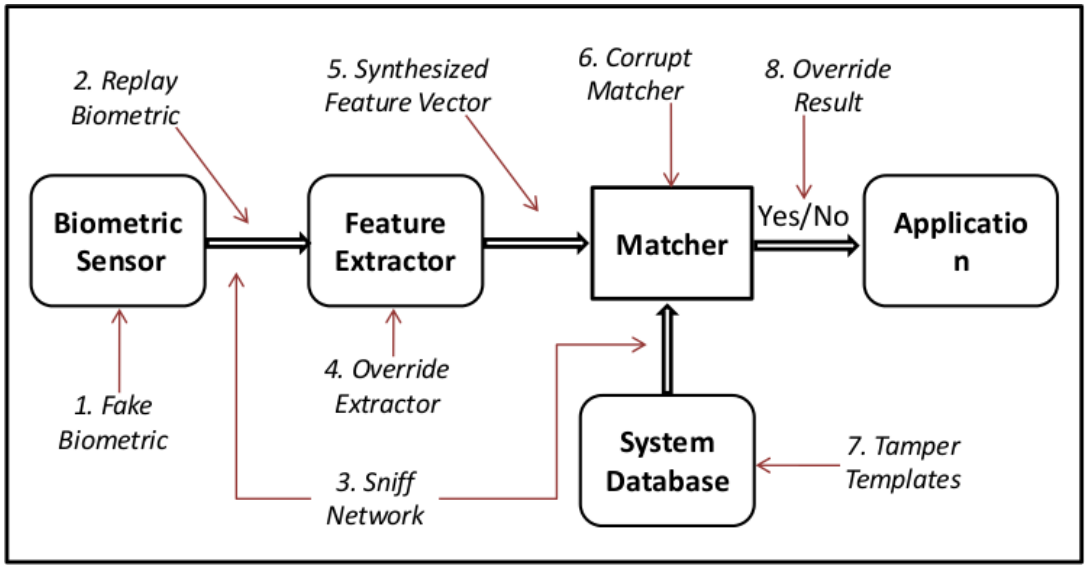

Figure 1.3 Attack model for a general biometric system.

– **Attacks on the network**

The interface between modules i.e. the network channel is susceptible to various attacks which may lead to leakage of sensitive biometric data and tampering of the information being transferred. A common active eavesdropping attack on a network is the *man-in-the-middle* attack. There, an adversary impersonates node endpoints to the satisfaction of another node and thus intercepts the information being transmitted. It allows the adversary to hijack a genuine user's session and gain illegitimate access to the system.

To avoid such attacks cryptographic encryption and signing protocols should be deployed to verify the data sent between the modules.

– **Attacks on the system software**

The software running on each module is susceptible to virus attacks, causing a dis-functioning or computing desired results when provided with certain inputs. A complete replacement of the software module is also a potential threat. The software systems may themselves have certain bugs such as providing a certain angle of face image or keeping two fingers on the sensor may always lead the system to declare a match. Such vulnerabilities, if exposed, form easy targets for intrusion.

To avoid such attacks the software modules must be built following secure code practices, they must be heavily tested for complete range of inputs and execution scenarios.

– **Attacks on the system storage**

The biometric template databases are susceptible to break-ins that compromise sensitive user biometric information.An adversary can replace the stored templates with fake ones or his own and open up the system for intrusion. The leaked templates can further be used to

generate biometric samples which in turn can be used to target other systems where those individuals might be enrolled.

To prevent such attacks the template database must have an extremely secure access. A large number of template protection techniques have been proposed in the literature to avoid breach of user privacy when the template database is compromised.

- **Biometric covertness** Biometric traits of individuals are not secret. Capturing pictures of a face, fingerprints, palmprints, voice samples and even iris covertly is possible using a camera or a microphone. The captured samples can then be used for intruding into a system where that individual enrolled. Such methods are useful for individual specific attacks. Gaining access to a large user biometric traits is generally possible through a direct attack on the database.

  Though it is difficult to avoid leakage of personal biometric traits, such attacks can be avoided through a multi-factor authentication process using smart-cards, passwords or PINs. The mass breach of biometric traits of users can be avoid by improving database security and implementing template protection mechanisms.

1.1.3 Effects of template compromise

Compromised biometric information can be used in multiple ways by an adversary. A spoof biometric sample can be reconstructed using the recovered template. The template may also be used for a replay attack for illegitimate access to a system and later, the stored original template may be replaced with a fake one to deny access to the legitimate user. Database linkage can be confirmed from the recovered biometric template. Adversary can track the activities of individuals enrolled with multiple servers. Different servers may contain different bits of personal information stored with them, the compromised biometric will act as a key to all of them allowing consolidation of all the information. Certain medical conditions about an individual can also be recovered from his biometric sample. This can be misused by the adversary as it may reveal confidential personal information.

Thus, protecting the biometric templates is of prime importance to prevent these consequences. Template protection techniques accompanied by mechanisms to detect biometric liveness helps to build a secure biometric authentication system.

1.2 Motivation and contribution

The area of biometric security is developing around these system vulnerabilities aiming to secure each of them with efficient protocols and without affecting the performance of the biometric system. Widely used cryptographic encryption schemes such as AES or RSA cannot be directly applied on biometrics as with passwords as the primary requirement of matching two biometric samples is to perform a fuzzy match. Cryptographic protocols are efficient at exact matching as in the case of passwords

or tokens. There exist several homomorphic encryption schemes which allow certain operations to be performed on encrypted data which can be extended to biometric systems. Several biometric template protection mechanisms have also been developed to permit secure biometric authentication. These mechanisms are usually specific to biometric traits and ways of feature representation. Ensuring perfection in combining both the aspects of security and credibility/accuracy and building a system efficient enough to work in real time has not yet reached a satisfactory stage. This is the source of our research motivation. In our work we focus on biometric verification systems where a one-one match is performed. Verification systems, as compared to the identification systems, are widely commercially used and operate in a relatively less controlled environment thus their implementations have more concerns of security.

We propose a novel biometric authentication protocol which performs secure, privacy preserving biometric authentication along with providing template protection for biometric traits which rely on normalized hamming distance as the dissimilarity measure. This protocol is based on a *somewhat* homomorphic encryption scheme and allows the use of biometric masks along with the feature vectors that hide the invalid values in the feature vector. Our experiments show that the authentication can be performed in real-time and permits linear scaling by increasing the number of processors.

We also propose one-time biometric tokens used for authentication in scenarios such as a bank transaction with some service provider. In such scenarios, individuals are forced to trust the service provider with their sole bank credentials. With one-time biometric tokens the trust condition can be relaxed as these tokens cannot be reused or used elsewhere. This protocol is based on the fuzzy commitment scheme which uses error correcting codes along with biometric feature vectors to permit secure authentication. This protocol also upholds the template protection and privacy requirements.

We have developed our protocols on binary biometric representations as they are efficient for storage and computations. Usually mis-matches in the binary representations i.e. a bit-wise XOR operation needs to be evaluated to compare two biometric samples, which proves to be computationally inexpensive than other matching techniques.

1.3 Organization of the thesis

In Chapter 2 we discuss the background of biometric security protocols, describe and analyze different template securing methods developed. We also discuss the use of cryptographic encryption methods used in relation to biometrics. In Chapter 3, we shall discuss existing cryptography based biometric authentication protocols and propose our secure hamming distance based authentication. In Chapter 4, we shall discuss existing time based secure authentication protocols and describe our one-time biometric token based authentication. In the above mentioned chapters, we shall also discuss the experiments carried out on some biometric traits and the performance results obtained.

Chapter 2

Background

Secure biometric authentication systems are developed to address the vulnerabilities discussed in the previous chapter. A secure authentication system must provide security to the biometric queries and templates and defend various attacks on the communication network. The main challenge is to handle all the security and performance requirements incorporating the intra-class variations in the biometric samples collected. Biometric features derived from two separate acquisitions of the same biometric trait are never exactly the same. So the widely used encryption schemes like AES or RSA cannot be used to store and match the templates, as even a slight difference in the data to be encrypted will result in large differences in their encryptions. Decryption of the template before a match to compute the score is not preferred as it leaves another vulnerability where sensitive biometric features can be sniffed by an adversary. These issues are addressed by methods which can be classified in two categories. The first is by using biometric template protection schemes. Several template protection schemes have been developed over the years which attempt to modify and store the biometric features in certain ways to keep the template secure. The authentication process also uses those modifications on the query and then computes the matching result. These template protection schemes are build keeping the following points in consideration. The templates created must be *1. Secure*, it must be computationally hard to recover the original biometric from the secure template. *2. Cancelable*, it must be possible to revoke an existing template in case of a compromise and issue a fresh one. *3.* Offer *multiplicity*, several templates of the same biometric must be diverse to ensure privacy. *4.* Offer acceptable *performance* in a real world application. The other category is based on the use of cryptographic protocols of homomorphic encryption or secure two-party computations which allow building secure privacy preserving matching protocols on top of a plain biometric system without degrading its performance. We discuss these different categories in the following sections using some standard schemes developed under each of them.

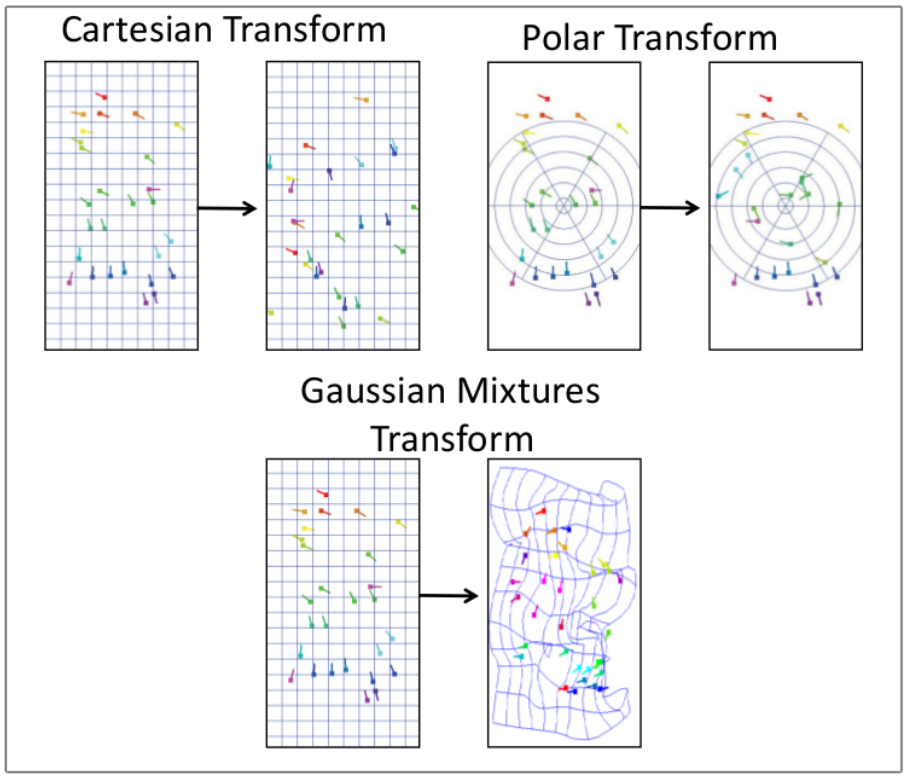

Figure 2.1 Some Non-Invertible Transforms applied to biometric features.

2.1 Template protection schemes

The template protection schemes [6, 67] are broadly classified into two categories - *Cancelable transforms* and *Biometric cryptosystems*. We discuss the details and some examples of the schemes in each category below.

2.1.1 Cancelable biometric transformations

These schemes are designed to hide the original biometric. A feature transformation function is applied to the biometric template and only the transformed template is stored in the database. The aim is to make the recovery of the original biometric difficult, also the transformed data must not reveal any details of the enrolled biometric. It must be ensured that the transformations handle the intra-class variations in the biometrics without affecting individuals' biometric characteristics. Such a system is cancelable as the transformations, if compromised, can be replaced using different transforms or different transform parameters. The same transformation is applied to the query features during the matching phase and then the transformed query is directly matched against the stored template.

Cancelable biometrics can be distinguished into two main categories:

2.1.1.1 Non-invertible transforms

As the name suggests, non-invertible functions are applied to transform the biometric features in this approach. An user specific random key defines the transformation parameters. Only the transformed features are stored in the database as templates. During authentication, if the transformation parameters and the biometric features match, the user can successfully generate a similar template. The non-invertible nature of transfoms ensures the safety of the biometric data even when the templates and the transform parameters are compromised. The intra-class variations are handled by the use of transformation functions tolerant to input variations.

The idea was introduced by Ratha *et al.* [64] as cancelable biometrics using non-invertible transform. He analyzed three transformation functions - cartesian, polar and surface-folding. For minutia based fingerprint matching, the minutia space is tessellated into a rectangular grid of cells and based on user specific keys those cells are mapped to the their new positions. As the minutia remain in the same space after transform, existing matcher can be used to match two fingerpints. In [63], Ratha *et al.* translate the minutia points using mixture of 2D gaussian functions and electric potential field randomized by charge distribution. User specific key defines the parameters of the weights, covariances, kernel centers and phase offset values in the case of gaussians and magnitude and position of charges in the case of electric potential field. As multiple minutia can get mapped to the same location, the non-invertibility property of the scheme is upheld. Figure 2.1 explain some of these non-invertible transforms. Various types of transforms to construct multiple cancelable biometrics from pre-aligned fingerprints and face images have been proposed in [62, 71, 74]. In [84], cancelable iris templates have been proposed where four different transforms applied to the iris feature domain, also reporting small drop in matching accuracy. A similar approach, adding row permutations to the set of functions, applied to iris-codes is suggested in [66].

Cryptographically secure biotokens have been proposed by Boult *et al.* [13, 14]. Biometric features are first transformed through scaling and translation. They are then split into a stable and an unstable part. The stable parts are secured using a cryptographically secure one-way transform and the unstable part is left in plaintext. Such secure biotokens have been proposed for face and fingerprints. In their fingerprint matching protocol, they secure the minutia descriptors by splitting the minutia position - co-ordinates and angle information into a stable integer part and a variable increment. Biotoken consists of the encrypted or hashed integer and the residue in plain. This approach provides provable template security as a strong one-way transform is used. During matching of a probe, similar transformation is applied to the probe. The stable part is expected to match exactly with the template, whereas the residue should fall within a matching window. To declare a biometric match, only a subset of the features are expected to perfectly match. Moreover, the matching accuracy is shown to be improved due to the applied transforms.

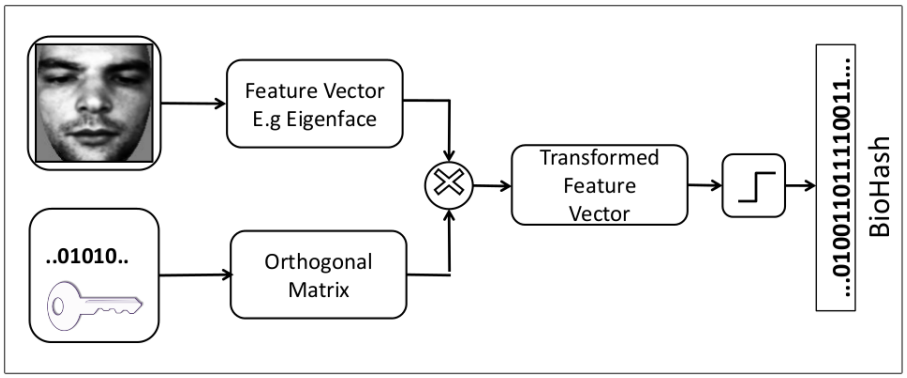

Figure 2.2 A Biohashing Scheme based on convolving the feature matrix onto a orthogonal matrix.

2.1.1.2 Biometric salting

In this approach, usually, invertible transforms using user specific keys are applied to the biometrics. The same transformation key must be provided by the user during each authentication attempt. The use of invertible transforms supports to maintain the matching performance of the biometric scheme. The keys used for transformation improve the entropy of the biometric features, thus hardening the recovery of original biometric from the template.

Some popular salting based methods are biohashing [72, 45], salting for face template protection [69]. Teoh *et al.* [73] introduced the biohashing scheme for face templates. Face features are processed with fisher discriminant analysis and then a random multi-space quantization is applied to them. The obtained result is subjected to a random orthogonal projection and further binarized based on a chosen threshold. User specific keys generate the projection vector providing the required security to the scheme. Iris based biohash has been proposed in [70] and palmprint based in [22]. Figure 2.2 explains an example of a Biohash.

To prevent impostors from attacking different databases and perform a cross-matching, different transforms are suggested for different applications.

2.1.2 Biometric cryptosystems

These systems are of two categories - key-generating and key-binding systems. The query and the template biometric are not directly matched in these systems, rather exact matching of generated or binded keys decides the authentication result. During enrollment *helper data* is computed using the biometric sample and a random key, which later assists in recreating the key based on the query biometric. The helper data must not reveal any significant information about the biometric. The two categories are classified based on the way in which the helper data is generated.

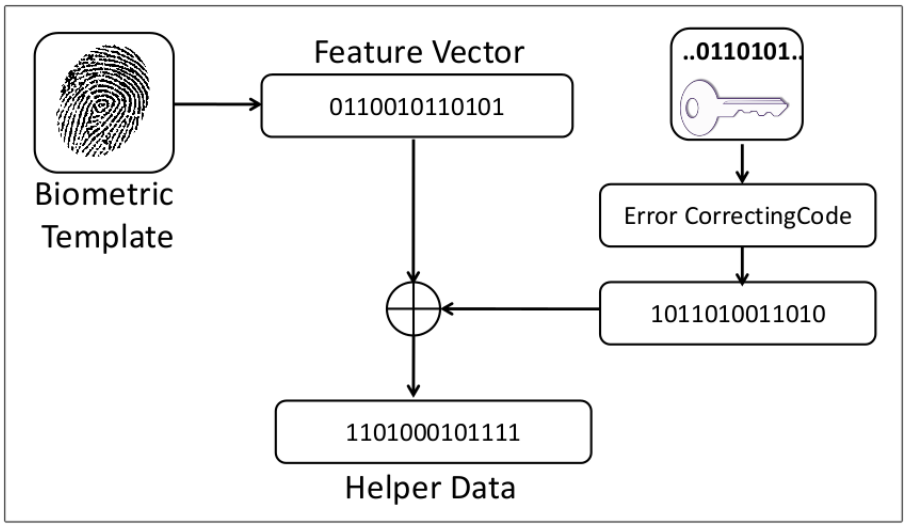

Figure 2.3 A Fuzzy Commitment Encryption Protocol.

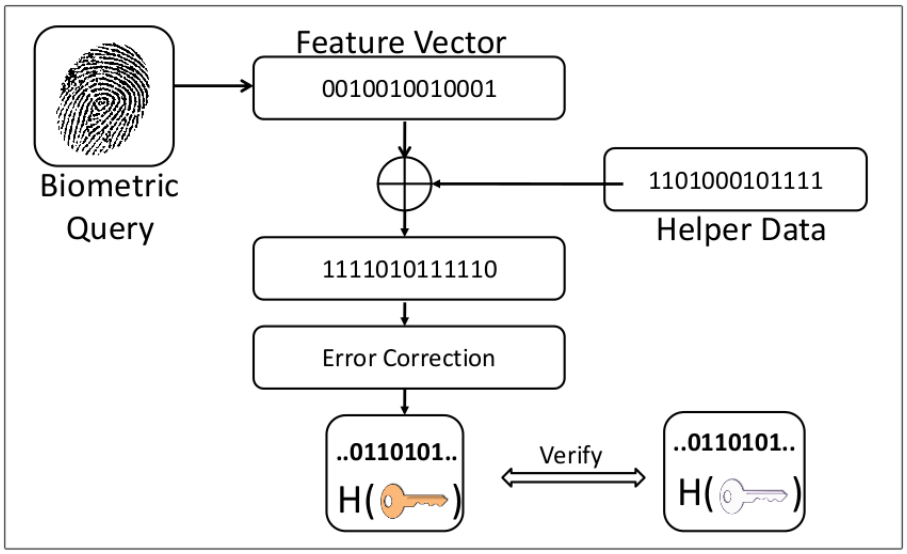

Figure 2.4 Fuzzy Commitment Decryption Protocol.

2.1.2.1 Key-generating systems

In these schemes the helper data is obtained only from the biometric sample. The authentication key is recovered from the helper data directly using the query biometric. Most of the key-generation schemes do not store the helper data; keys are directly generated directly from the biometric data. This has an impact on the performance of such systems as the intra-class variations in biometric acquisitions make the task difficult. Also, if the keys are generated only from the biometric features then the revocability of the scheme is affected.

Chang *et al.* [20] proposed a key-generation scheme from face biometrics. They perform user-specific biometric feature quantization and store the boundary information as helper data. The quantization bins are created using distribution values of the user's feature vector. The intra-class variations caused during authentication are thus handled through the helper data. Key-generation schemes of *fuzzy extractor* and *secure sketch* were proposed by Dodis *et al.* [28, 27]. Iris based key-generation was proposed by Davida *et al.* [26] where the hash of the template is the extracted key. The helper data stored is used for error correction of extracted query templates. Multiple acquisitions of the iris are considered and a majority voting scheme is used to improve stability of the bits of the template. These bits are appended by error correcting bits and a hash is computed to obtain the key.

Low key stability and entropy are common with these schemes as the attempt is to directly extract keys from the biometric. The concerns of repeatedly generating the same key or generating different keys for different services are not addressed satisfactorily usually in these schemes.

2.1.2.2 Key-binding systems

In these schemes the helper data is obtained by binding a chosen key to a biometric template. By applying an appropriate key retrieval algorithm, keys are obtained from the helper data during authentication. The cryptographic keys used are independent of the biometric features allowing construction of revocable templates. Updating the template with a different key requires a re-enrollment.

One of the widely used work amongst the key-binding schemes is the fuzzy commitment scheme developed by Juels and Wattenberg [47]. It uses the technique of error correcting codes accompanied by binary operations to develop systems with cryptographic security standards. During registration a fixed-length feature vector is extracted from the user's biometric, let it be B. A random key K is encoded using an error-correction code into a codeword W. Both B and W are binary vectors, their bitwise XOR $T = B \oplus W$ is stored along with the cryptographic hash $H(K)$ as the helper data. At the verification stage, using the user's biometric sample, a feature vector B' is computed. The authenticator computes $B' \oplus T$, the user's template in the database, to obtain $W' = W \oplus e$, where $e = B' \oplus B$ i.e. the difference between the query and the template biometric. If a valid biometric has been provided the $1\ bits$ in e must be sufficiently few. W' on decoding outputs K', which is hashed to obtain $H(K')$. If the errors are within the error-correction capacity of the codeword the exact initial key K is recovered. That is verified

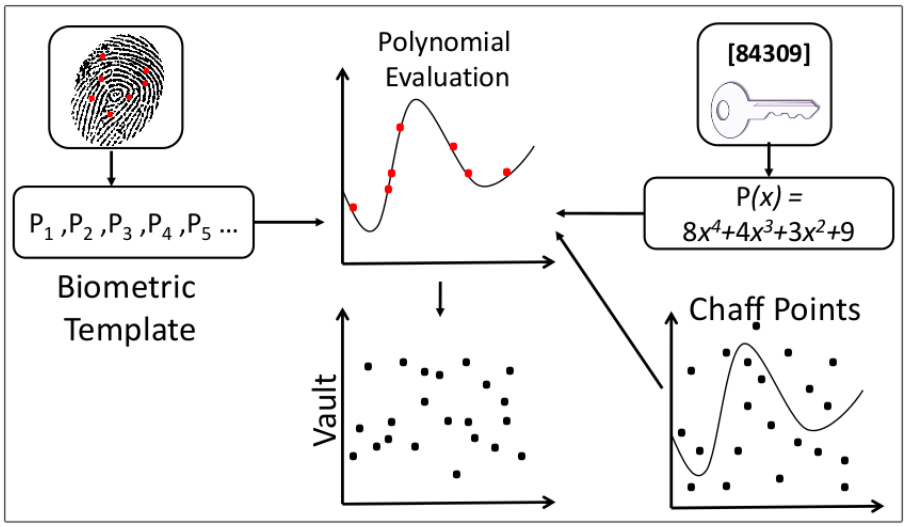

Figure 2.5 Fuzzy Vault Encryption Protocol.

by matching $H(K')$ and $H(K)$. A successful match indicates a successful authentication. Figure 2.3 and 2.4 describe the fuzzy commitment protocol.

Veen *et al.* [80] propose fuzzy commitment scheme for face biometrics. First a binary quantizer is applied on the face features followed by a reliable bit selection process based on stastitical values to output a fixed length vector for commitment. Nandakumar *et al.* [55] propose a scheme based on binarized phase spectrum for fingerprints. A fixed length binary string obtained by quantizing Fourier phase spectrum of fingerprint minutia is secured through fuzzy commitment using trubo codes. The minutia are first aligned based on the focal point of high curvature regions. Tong *et al.* [75] propose a key-binding scheme based on stable and ordered representation of fingerprints i.e. the fingercode. Hao *et al.* [38] applies the fuzzy commitment scheme to iris-codes. Cryptographic keys prepared with Hadamard and Reed-Solomon error correction codes are binded with the 2048 bit iris-codes. The results were impressive which were not achieved till then.

Fuzzy vault is another popular key-binding scheme introduced by Juels and Sudan [46]. It usually applies on an unordered set of biometric feature values to construct a fixed length key. If a considerably similar set is produced during authentication the same fixed key will be recovered. In case of fingerprints, the minutia co-ordinates extracted form the unordered set P and a fixed key K used as coefficients of a polynomial provides the encoding function. The minutia values are evaluated on the polynomial to create the vault V. Chaff points are added to the vault to hide the genuine values. Also, a hash of the key $H(K)$ is stored along with the vault. During authentication, the minutia co-ordinates of the query are matched with the values in the vault and the corresponding polynomial values are decoded using Lagrange interpolation to obtain a polynomial P'. In order to correctly decode the vault, sufficient elements of the query must overlap with genuine elements in the vault. The key obtained K' is hashed and verified with the stored $H(K)$. A successful match indicates a successful authentication. Additional

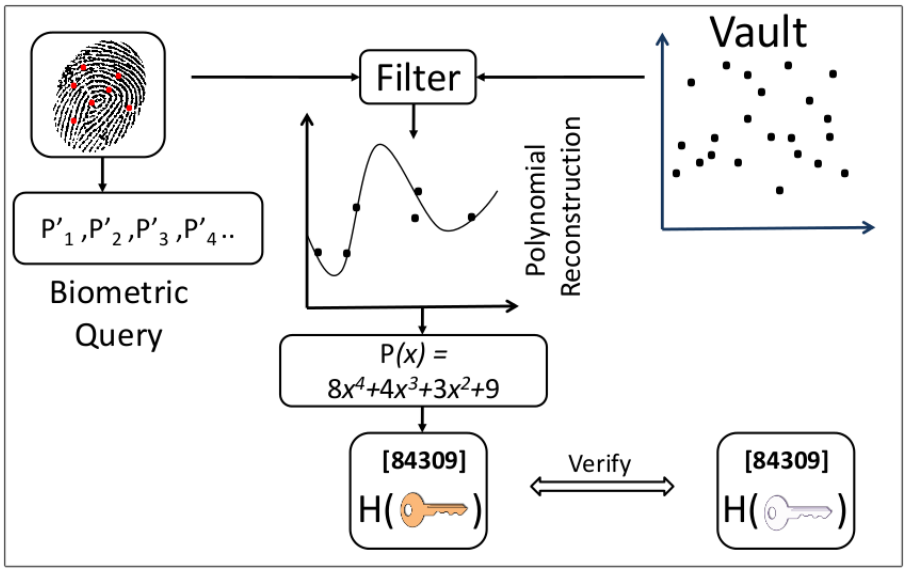

Figure 2.6 Fuzzy Vault Decryption Protocol.

layer of error correction can be added before key generation to compensate for small differences in key recovery. Figure 2.5 and 2.6 describe the fuzzy vault scheme.

Based on the Juels and Sudan's proposal, a practical implementation of a fingerprint vault was provided by Clancy *et al.* [21]. The vault is created using the averaged feature values of multiple fingerprint instances. The problem of alignment of fingerprints is not addressed by them. Nandakumar *et al.* [57] propose a method for aligning the query and the template based on helper data computed from high curvature points derived from fingerprint orientation field and use that prior to matching minutia in the fuzzy vault. Dodis *et al.* [28] propose an improvement to [46] by using a high-degree polynomial instead of chaff points. Wu *et al.* [82] propose a fuzzy vault for face based on binarized PCA(Principal Component Analysis) features and supported by Reed-Solomon error correction. Fuzzy vaults for iris biometrics is proposed by Lee *et al.* [52]. Independent component analysis(ICA) is performed on the iris image and the obtained representative features of multiple iris samples are clustered to overcome the variance. The cluster centers are then used to construct the fuzzy vault. Some other fuzzy vault schemes are present in [78, 52, 77, 43].

2.2 Cryptographic protocols

The following cryptographic protocols have been leveraged to build secure privacy preserving authentication systems. They provide significant advantage in security of the biometrics but the application feasibility must supported with reasonable computation costs.

Template Protection Scheme	*Advantages*	*Limitations*
Non-Invertible Transform	Hard to recover original biometric, Multiplicity through user specific functions	Non-invertibility v/s Accuracy trade-off
Salting Schemes	Multiplicity and lower FAR through user-specific keys	Handling large intra-class variations
Key-Binding Schemes	Accommodate intra-class variations, Provide acceptable security	Trait specific error correction selected, Helper data must be secure
Key-Generation Schemes	Keys extracted directly from the template	Low Discriminability, High key Stability v/s Entropy trade-off

Table 2.1 Comparison of various template protections schemes, their advantages and limitations.

2.2.1 Two-party computation

Secure two-party computations can be adopted to address the problem of secure biometric authentication. The client side wishes to get its biometric query verified by the server without revealing it. Several protocol have been built using garbled circuits and oblivious transfer which offer privacy of individual data when the other party securely evaluates a function on it. These protocols consider a threat model with *semi-honest* parties i.e. the parties are expected to follow the protocol honestly but they may try to acquire additional information about other party's private data through intermediate results available with it. Such protocols can be modified to uphold stronger security measures through generic ways [53].

In a general framework, the client has some input x, the server has input y and they wish to compute a function $f(x, y)$ without leaking one's values to the other. Only the output of the function should be revealed. The function f is expressed in the form of a boolean circuit accepting binary converted x and y. One party computes an "encrypted" version of the boolean circuit, also called a garbled circuit and provides it to the other party. On receiving the circuit, the other party then obtains the keys corresponding to its actual input bits using oblivious transfer and then evaluates the circuit. The result can be revealed to one or both the parties depending on the implementation. Oblivious transfer allows a sender, having two values, to transfer to a receiver exactly one value of his choice without leaking any information about the other. Also, the sender does not get to know which value the receiver had requested for. We limit the discussion on workings of garbled circuits as we do not use them in any of our protocols. Detailed practical implementation of garbled circuits with oblivious transfer is present in [61].

2.2.2 Homomorphic encryption

An encryption scheme which allows certain algebraic operations to be performed on the plain data by performing specific operations on the corresponding ciphertext is an homomorphic encryption scheme. This enables users to allow processing of their biometric features without exposing the underlying plaintext features. Let $E(x)$ and $E(y)$ be the encryptions of two values x and y respectively. A homomorphic

encryption scheme, for a $\odot$ operator applicable to plaintexts and for a $\oplus$ operator applicable to ciphertexts, the following holds true:

$$E(x) \oplus E(y) = E(x \odot y) \tag{2.1}$$

A fully homomorphic encryption scheme allows to compute addition and multiplication homomorphic operations which in-turn can be used to perform any polynomial computations on the ciphertexts. Introduced by Gentry *et al.* [36] and followed by several improvements and implementations, efficient implementations to be used in real-time applications are not yet available. A partially homomorphic encryption scheme allows only one operation to be performed homomorphically. There are several partially homomorphic encryption protocols - *Paillier* allows for separate additive and multiplicative homomorphic encryptions, *Goldwasser-Micali* provides a homomorphic XOR operation, *ElGamal* cryptosystem provides a multiplicative homomorphic scheme.

Secure biometric matching protocols based on homomorphic encryption were introduced in [29] based on Paillier encryption and DGK encryption. [68, 30, 58, 9] use a combination of homomorphic encryption and garbled circuits. The protocols in [19, 30] just use garbled circuits, [8] uses Paillier and ElGamal cryptosystem, [18] uses the Goldwasser-Micali cryptosystem. [79, 18, 15] provide for secure storage of templates. [18, 15] support privacy of user's identity while authentication using private information retrieval protocols.

2.3 Summary

In this chapter we discuss several mechanisms in the literature to develop secure biometric systems and highlight their strengths and limitations. The choice of a particular scheme depends on the selected biometric trait, the type of features extracted and the amount fuzzy-ness involved. Non-invertible transforms can be used when using minutia co-ordinates as fingerprint features, whereas fuzzy commitment scheme would be preferred when used fixed length binary minutia descriptors. Fuzzy vaults with error correction would be preferred over non-invertible transforms when high intra-class variations are to be considered. Cryptography based schemes would be preferred over salting schemes when extremely high accuracy is expected and the time constraints relaxed.

In our work we have focused on fixed length binary representations and built secure authentication protocols for them. In the work discussed in Chapter 3, we use a cryptography based scheme which is *somewhat* homomorphic i.e. it additively homomorphic along with supporting one multiplicatively homomorphic operation to build a secure authentication protocol. Further, in Chapter 4, we adopt a scheme similar to the fuzzy commitment mechanism to build a one-time password based authentication protocol. The advantage of building such schemes is that they can be to extend to any biometric trait that chooses its feature representation as a fixed length binary vector.

Chapter 3

Secure Hamming Distance Based Biometric Authentication

3.1 Introduction and related work

In this chapter, we design and implement a novel biometric system which provides secure privacy preserving authentication mechanism along with template protection. It also allows the use of binary masks along with the biometric templates to hide invalid locations in their feature vectors. The proposed protocol requires the underlying matching algorithm to be based on normalized hamming distance of two binary feature vectors. It's a two party protocol with the server having the secure biometric templates and the client attempting the authentication using its own biometric data along with some keys. The authentication process takes place on encrypted data and does not allow any leakage of information about the biometric features. Various protocols to address important authentication issues have been developed earlier but they simultaneously do not support template protection and efficient computability while using masks together with security and privacy.

Secure privacy preserving face recognition was initially proposed by Erkin *et al.* [29] based on the cryptographic primitives of Paillier encryption and DGK cryptosystem. They provide a protocol for secure matching based on the *eigen-faces* face recognition algorithm. Additive homomorphic encryption schemes are used to compute the euclidean distance between vectors. The Paillier cryptosystem is used to calculate the euclidean distance and the DGK cryptosystem is optimized to perform a secure comparison. There exist limitations on the scalability of their system due to the high computation values even for small feature vector sizes.

Secure face matching protocols were also proposed in [30, 44, 58, 68]. Sadeghi *et al.* [68] improved the efficiency of Erkin's work by proposing a hybrid protocol that also uses homomorphic encryption to compute the euclidean distance, whereas it uses garbled circuits to perform a comparison(to find minimum score). They devised an idea of *packing* to save the communication cost of the protocol. They use the garbled circuits generated using generic compiler FairplaySPF [60], which in turn is based on Fairplay [54], which cannot develop efficient custom circuits. Efficient biometric application specific garbled circuits were developed in [30]. They too compute the euclidean distances using homomorphic encryption, but they apply the concept of *packing* from [68] to optimize the encrypted computations.

They propose a efficient backtracking protocol to convert the circuit's final output wire labels to meaningful indexes, obliviously transmitting information about the match.

SCiFi [58] is a practical privacy-preserving face identification system. It uses a component-based face identification technique with a binary index into a vocabulary representation. The matching score between faces is calculated using the hamming distance between their bit-vectors. The design uses homomorphic encryption and oblivious transfer for distance calculation and comparison respectively.

A work by Barni *et al.* [8] designs a privacy-preserving protocol for fingerprint identification using fingercodes [41]. They too use homomorphic cryptosystems for privacy preservation, specifically the additive ElGamal(over Elliptic Curves) cryptosystem to compute a *bit-MIN* comparison operation to decide the match result.

A secure iris and fingerprint matching protocol was proposed by Blanton *et al.* [9]. Similar to earlier protocols their work also uses a combination of homomorphic encryption schemes and garbled circuits for secrecy and privacy. The difference is the underlying biometric protocols use hamming distance as a dissimilarity measure for evaluating the samples. Iris matching is performed based on the iriscode technique and fingerprints are matched using the fingercode representation. They use the DGK homomorphic scheme, which is efficient for binary representations, provide optimizations to the garbled circuits and also perform some offline computations to reduce computational costs. Bringer *et al.* [19] propose a secure iris matching protocol just using garbled circuits. They incorporate a filtering technique, as proposed in [39], in their matching system. For a query sample produced, the filtering mechanism outputs a candidate set of possible template instances that could match with it. It is based on a beacon guided search which matches short beacons generated from the reliable bits of the feature vector to create the candidate set. Then a complete match based on the hamming distance is preformed on the candidates to obtain the result. Their solution significantly speeds up the identification process.

Bringer *et al.* [18] propose a secure privacy-preserving protocol based on Goldwasser-Micali cryptosystem. They consider the template database a different entity than the service provider and propose a private information retrieval(PIR) in order to ensure biometric privacy of the user from the service provider. PIR is also used in the [15] to propose a secure privacy preserving biometric authentication based on Secure Sketches. They use double encryption technique with Goldwasser-Micali scheme on the top of Paillier scheme. The security is based on a strong assumption that the service provider must be equipped with a Hardware Security Module(HSM) with which the secret keys are stored. The HSM performs the decryption, declares the result based on the verification of recovered sketch with the stored key hash.

Bringer *et al.* [17] address the problem of biometric identification in an encrypted domain. The combine Bloom Filters with Locality Sensitive Hashing to develop a error-tolerant search scheme. They apply that search mechanism in a hamming space to achieve identification. The result is a small candidate set of possible matches for which another secure matching protocol should be used. Its computational complexity is high for a real time authentication system. Though they provide a solution for

finding close matches securely, the further problem of identifying a correct match cannot be solved in a completely encrypted domain.

These above discussed protocols do not provide security, privacy, template protection and usability of masks with efficient computations simultaneously. [29, 68, 8, 58, 9, 19] provide secure privacy preserving protocols, but have the templates stored in plaintext. [18, 15] but do not involve masks. [17] operates in encrypted domain but outputs close matching set of vectors, exact matching computational costs have not been discussed.

A system in totally encrypted domain which provides secure and private authentication is proposed by Upmanyu *et al.* [79]. Its design is limited to biometric schemes which use linear classifiers and SVMs. It is based on the Paillier cryptosystem and utilizes its additive and multiplicative homomorphic properties by changing the encryption scheme during the multiple rounds of communication. The intermediate values are randomized to maintain privacy on decryption during change of schemes. Just final matching score is revealed to the authentication server to decide the result.

Biometric systems for fingerprints, iris as well as palmprints exist which are based on hamming distance as the dissimilarity measure. We limit the description of our protocol to iris matching based on iriscodes, but conduct experiments on iris and palmprint matching. The palmprint matching method, based on palmcodes, is similar to the iris matching technique. The proposed protocol ensures privacy, security, fixed rounds of communications for the authentication keeping intact the accuracy provided by the underlying matching algorithm.

3.2 Preliminaries

In this section we give an overview of the algorithms used by our protocol. We first describe the encryption mechanism and supported homomorphic operations. Then we briefly describe the Iris matching technique used.

3.3 Homomorphic encryption

An encryption scheme which allows certain algebraic operations to be performed on the encrypted data by performing certain operations on the corresponding ciphertext is an homomorphic encryption scheme. There are several partially homomorphic encryption protocols(*Paillier, Goldwasser-Micali, ElGamal*) which allow either addition or multiplication operations on the encrypted plaintexts. Protocols have also been proposed which allow fully homomorphic - addition and multiplication operations among encrypted plaintexts, but they are not efficient enough to be used for real time applications. We choose a scheme which is *somewhat* homomorphic i.e. it additively homomorphic along with supporting one multiplicativly homomorphic operation. This suits our application as only one homomorphic multiplication operation is sufficient for establishing the security for the protocol.

3.3.1 BGN cryptosystem

We use the *somewhat* homomorphic encryption scheme proposed by Boneh [12] constructed on the lines of the Paillier's encryption scheme [59]. It allows additive homomorphism of ciphertexts, also supports *one* multiplication operation between the ciphertexts. The cryptosystem is based on finite composite order groups built on elliptic curves that support bilinear maps. Its security depends on the hardness of the *subgroup decision problem* - the problem of determining whether a given element of a finite group G lies in a specified proper subgroup G_1 or not. Then for composite order groups, say of order $n = p_1p_2$, the hardness of the subgroup decision problem directly transforms to the hardness of factoring n. High security demands that it must be infeasible to factor it, so n must be considerably large. The drawback is - computing group operations and pairings on large composite order groups is prohibitively slow. Prime order groups on the other hand can provide equivalent security with a smaller order[7] and allow faster computation of pairings. The subgroup decision problem can be constructed on a prime order group using a random generator and a subgroup. The generator of the subgroup is the group generator raised to a random integer in its field. A Tate pairing on a $1024\ bit$ composite order elliptic curve group is roughly 50 times slower than that on a comparable $160\ bit$ prime-order curve group[32].

Freeman[32] proposed a system to convert composite-order groups to prime-order that encompasses the bilinear properties and holds the hardness of the subgroup decision problem. It provides a framework for using prime-order elliptic curves to construct bilinear groups to create efficient versions of the cryptosystems that originally used composite-order groups. The generated prime-order groups are equipped with projection-maps that map them onto proper subgroups and commute with the pairing. We implement this particular scheme and use it to build our protocol. The prime-order groups also support the required homomorphic additions of ciphertexts along with *one* multiplication. The algorithms provided by the scheme are described below.

KeyGen(λ): Let $\mathcal{G}$ be a projecting bilinear group generator. Provided a security parameter λ, compute $(G, G_1, H, H_1, G_t, G'_t, e, \pi_1, \pi_2, \pi_t) \leftarrow \mathcal{G}(\lambda)$. Choose $g \xleftarrow{R} G, h \xleftarrow{R} H$ and output the public key $PK = (G, G_1, H, H_1, G_t, G'_t, e, g, h)$ and the secret key $SK = (\pi_1, \pi_2, \pi_t)$. *Where*,

λ is the security parameter, to generate $\lambda\ bit$ prime p.

G, G_1, H, H_1, G_t and G'_t are Groups of order p over an elliptic curve such that $G_1 \subset G, H_1 \subset H\ and\ G'_t \subset G_t$.

g, h are random generators of G, H respectively.

$e = G \times H \rightarrow G_t$ is a bilinear map.

$\{\pi_1, \pi_2, \pi_t\}$ are the projection maps of their corresponding groups, trapdoors used in the decryption process.

Encrypt($\mathcal{PK}, m$): To encrypt a message m using a public key, choose $g_1 \xleftarrow{R} G_1$ and $h_1 \xleftarrow{R} H_1$ and output the ciphertexts $(C_g, C_h) = (g^m.g_1, h^m.h_1) \in G \times H$.

Decrypt($\mathcal{SK}, \mathcal{C}$): The input ciphertext C can be an element of $G, H\ or\ G_t$.

- If $C \in G$, output $m \longleftarrow log_{\pi_1(g)}(\pi_1(C))$

- If $C \in H$, output $m \longleftarrow log_{\pi_2(h)}(\pi_2(C))$
- If $C \in G_t$, output $m \longleftarrow log_{\pi_t(e(g,h))}(\pi_t(C))$

Homomorphic properties: The system is additively homomorphic within the respective groups and supports *one* multiplication operation between the groups. So, ciphertexts in G are additively homomorphic as the ciphertexts in H and G_t. To obtain a multiplicative homomorphic operation, one ciphertext must be from G and H each. A homomorphic multiplication will result to a ciphertext belonging to G_t. There is no reverse mapping i.e. mapping from G_t to $G \times H$ without the private key. This also shows the limitation of the scheme, that these three groups allow only one such multiplication. Details on individual operations are mentioned below.

Add$(\mathcal{PK}, \mathcal{C}, \mathcal{C}')$***:*** The two ciphertexts C, C' are in one of G, H *or* G_t. Choose $g_1 \xleftarrow{R} G_1$ and $h_1 \xleftarrow{R} H_1$ and perform:

- If $C, C' \in G$, output $C.C'.g_1 \in G$
- If $C, C' \in H$, output $C.C'.h_1 \in H$
- If $C, C' \in G_t$, output $C.C'.e(g, h_1).e(g_1, h) \in G_t$

Along with addition of two ciphertexts, it also permits multiplication of a ciphertext with an integer in plaintext. This is an iterative operation where the ciphertext is added to itself the necessary number of times to get the required product.

We use $+_E$ symbol for this operation in our protocol.

Multiply$(\mathcal{PK}, \mathcal{C}, \mathcal{C}')$***:*** The two ciphertexts are - $C \in G$ and $C' \in H$. Choose $g_1 \xleftarrow{R} G_1$ and $h_1 \xleftarrow{R} H_1$ and output

- $C = e(C, C').e(g, h_1).e(g_1, h) \in G_t$

We use $*_E$ symbol for this operation in our protocol.

One can perform the above operations on the specified pair of ciphertexts without getting any details about the underlying encrypted message. Also, the resulted ciphertexts do not reveal any information about the ciphertexts involved in the operation.

3.4 Iris matching

The iris feature extraction algorithms [81, 23] process the scanned iris image and compute a binary vector, let it be B with length $k\ bits$. The processing involves applying various Gabor filters on separate local areas of the iris image to generate an iriscode. The feature extraction algorithm, along with the feature vector, outputs the reliable and unreliable bits in it. This is provided through a binary mask vector of same length, with a '1' at the location which is valid or reliable and '0' at the position where the value in the feature vector should be discarded. So, provided two biometric feature vectors B' and

B and their corresponding masks M' and M, the normalized hamming distance between them is used as a measure of dissimilarity between the two samples. The normalized hamming distance is given by

$$d_H(B', M', B, M) = \frac{||(B' \oplus B) \cdot M' \cdot M||}{||M' \cdot M||} \tag{3.1}$$

The distances are calculated over some rotations of the second template as the two images can be slightly misaligned. By considering the minimum of the distances calculated, the minor orientation errors are compensated. A threshold value τ is chosen upon by the matching algorithm and based on the comparison of d_H with τ, the result is a match or a mis-match. This τ is based on the distributions of authentic and imposter data. It is selected in a way to achieve the desired FAR and GRR.

Palmcode based palmprint matching is very similar, also uses the same dissimilarity measure[83]. The other matching techniques involving extracting principal lines from the palmprint [50] lack desired accuracy. One of the reasons with them is unclear wrinkles in certain palmprints. The circular Gabor filter, effective for texture analysis, proves useful with palmprints in extracting the reliable palmcode vectors. During the matching phase, to accommodate improper processing and slight orientation differences, one of the feature vector is translated and matched in certain small range. As with iris matching, the minimum distance value obtained is the final matching score. We perform our experiments on these two biometric modalities.

3.5 Proposed protocol

The proposed system is a two phase protocol, built over a *client-server* architecture. Initially, the user registers himself on the server, whose access he seeks, by submitting his biometric data to it. This is the enrollment phase. The user extracts and sends its biometric features, in our case the iris template(feature vector and mask) to a trusted enrollment server. The server produces two binary keys, one for the iris feature vector and the other for the mask. The values stored on the server are the XORed values of the template vector with the corresponding key. Let B be the binary iris feature vector and M be its mask provided by the user. The server generates two bit vectors K_b and K_m of lengths equal to that of B and M respectively. The server stores $(B \oplus K_b)$ and $(M \oplus K_m)$ on it and the keys K_b and K_m are handed over to the user through a smart card. They will be required while authenticating himself in the future.

In the authentication phase the user, client from now on, has to prove its identity. It gets involved in a protocol with the server for gaining access to the application built on it. It is a three round protocol with the client extracting his biometric information and sending only the encrypted version to the server. The server only operates on ciphertexts over the protocol to finally obtain the authentication score of the client biometric. Based on the score, it then sends just the authentication result to the client. *Figure 3.1* gives the overview of the protocol. We describe it below in detail.

Authentication phase:

The client extracts the feature vector B', mask M' from its iris biometric. It then uses its smart card to extract the two keys, K_b and K_m. It then generates a pair of public and private keys of the

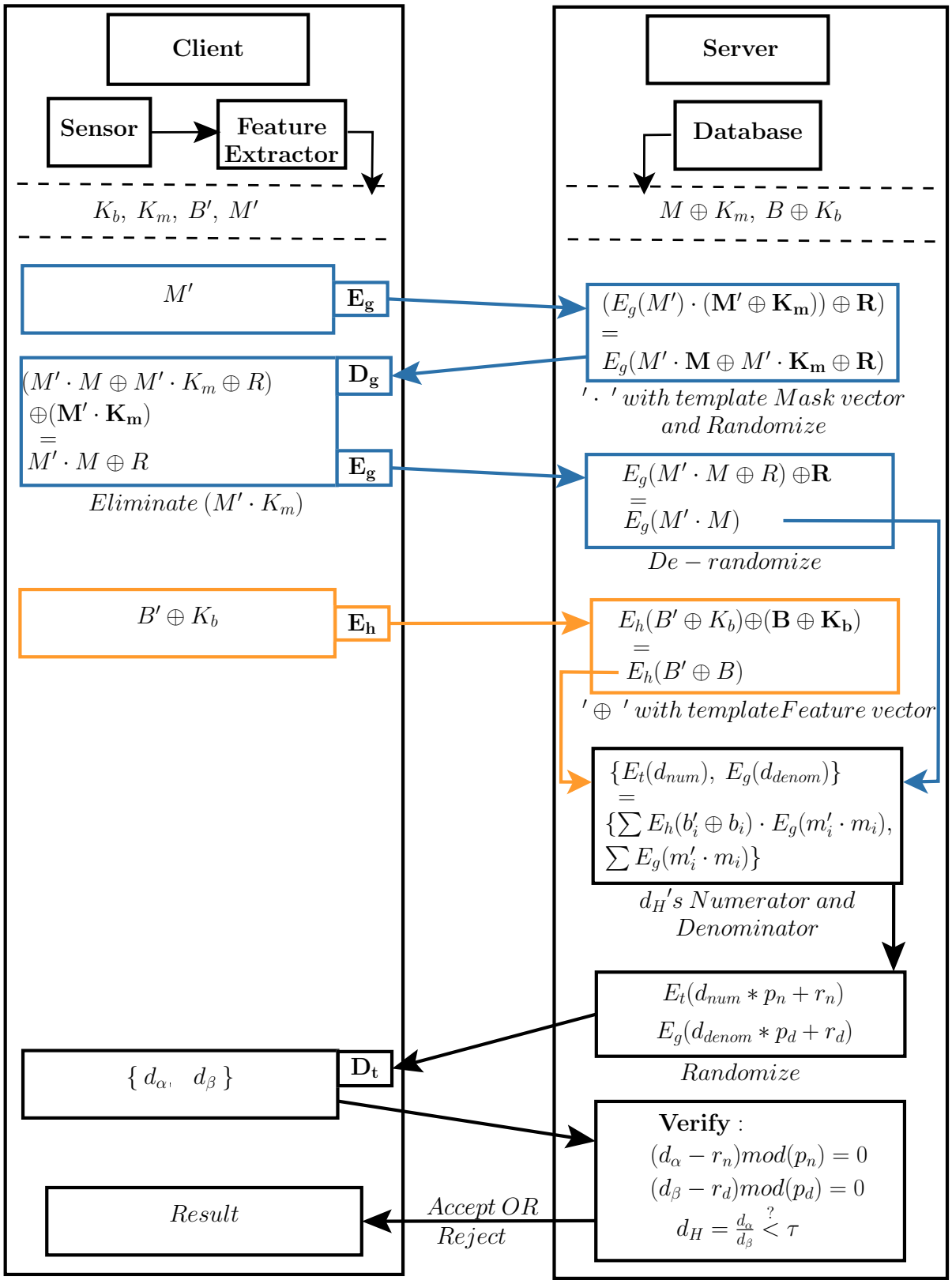

Figure 3.1 Protocol for secure hamming distance based biometric authentication using masks.

above mentioned cryptosystem. The public key $PK = (G, G_1, H, H_1, G_t, G'_t, e, g, h)$ and the secret key $SK = (\pi_1, \pi_2, \pi_t)$. The client publishes his public key to the server and then follows the protocol described below.

INPUT:

The client's input is - biometric vector B', biometric-key K_b, mask M' and mask-key K_m each $\in \{0,1\}^l$. The server has the enrolled biometric templates $B \oplus K_b, M \oplus K_m$ of all the users registered on it and a threshold $\tau = \tau_n/\tau_d$.

OUTPUT:

The server learns α(matching score) of the client's biometric template and then communicates the authentication result to the client.

PROTOCOL:

1. The client performs the G group encryption on each bit of M' and sends it to the server along with it's identity.

2. The server receives $ID, \{E_g(m'_0), \ldots, E_g(m'_{l-1})\}$, generates a random binary vector $R = \{r_0, r_1, \ldots, r_{l-1}\}$ and performs $\forall i, (E_g(m'_i) \cdot (m_i \oplus k_{mi})) \oplus r_i$ to get the following values:

   $\forall i, (E_g(m'_i \cdot m_i \oplus m'_i \cdot k_{mi} \oplus r_i))$. It sends them over to the client. (Efficient ways of calculating the $\oplus$ and $\cdot$ operations of an encrypted value with a plain value are mentioned in the text below.)

3. The client decrypts the binary vector and eliminates the term involving the mask-key.

   $\forall i, (m'_i \cdot m_i \oplus m'_i \cdot k_{mi} \oplus r_i) \oplus (m'_i \cdot k_{mi}) = (m'_i \cdot m_i \oplus r_i)$

   It again performs G group encryption of the result and sends it to the server.

4. The server now receives $\{E_g(m'_0 \cdot m_0 \oplus r_0), \ldots, E_g(m'_{l-1} \cdot m_{l-1} \oplus r_{l-1})\}$ and calculates $\forall i, E_g(m'_i \cdot m_i \oplus r_i) \oplus r_i$ to get the final mask vector $\{E_g(m'_0 \cdot m_0), \ldots, E_g(m'_{l-1} \cdot m_{l-1})\}$

5. The client calculates $B' \oplus K_b$ and sends the H group encryption of its each bit to the server.

6. The server receives the encrypted bits and calculates $\forall i, E_h(b'_i \oplus k_{bi}) \oplus (b_i \oplus k_{bi})$ to get $\{E_h(b'_0 \oplus b_0), \ldots, E_h(b'_{l-1} \oplus b_{l-1})\}$

   Now, the server calculates

   $\forall i, E_t((b'_i \oplus b_i) \cdot m'_i \cdot m_i) = E_h(b'_i \oplus b_i) *_E E_g(m'_i \cdot m_i)$.

   It then calculates,

   $E_t(d_{Hn}) = +_E \sum_{i=0}^{l-1} E_t((b'_i \oplus b_i) \cdot m'_i \cdot m_i)$

   $E_g(d_{Hd}) = +_E \sum_{i=0}^{l-1} E_g(m'_i \cdot m_i)$

7. It then calculates $E_t(\tau_d \times d_{Hn}),\ E_g(\tau_n \times d_{Hd})$ using $+_E$ and performs $E_g(\tau_n \times d_{Hd}) *_E E_h(1)$ to obtain $E_t(\tau_n \times d_{Hd})$.

   It then computes $E_t(\alpha) = E_t(2048 + \tau_d \times d_{Hn} - \tau_n \times d_{Hd})$ using the $+_E$.

   Where, α will be in the range $[0, 4096]$. The server then randomizes it - $E_t(\alpha \times p + r)$ and sends it to the client.

8. The client decrypts the ciphertext and sends back the plaintext. Let it be β.

9. The server receives β and performs the de-randomization with a check. The server checks if $(\beta - r) mod(p) = 0$. If it holds then it sends the authentication result:

   $$Result = \begin{cases} 1 & \textit{If } (\beta - r)/p < 2048 \\ 0 & \textit{otherwise} \end{cases}$$

The client performs the encryptions as mentioned in *Section 3.3.1* $E_g, E_h\ and\ E_t$ denote $G, H\ and\ G_t$ group encryptions, which are additively homomorphic in their respective groups. A multiplication operation performed between two ciphertexts, one belonging to G and the other belonging to H outputs a ciphertext belonging to G_t. The client encrypts the bits of mask M' with E_g, whereas that of B' with E_h.

The server performs the required XOR and AND operations on individual group elements with known plain text bits as in *Steps 2, 4* and *6* of the protocol. Let $E(b'_i)$ be an encrypted bit, at location i of some vector available with the server. Let b_i be the server's own bit at location i. Server performs the '$\oplus$' operation in the following way. If $b_i = 0$, the result value is kept as $E(b'_i)$ itself. If $b_i = 1$, the server first negates the encrypted bit $E(b'_i)$ available with it to obtain $E(-b'_i)$ by performing a group inverse operation. Then $E(-b'_i) + E(1)$ is computed as the result.

For a '$\cdot$' operation with the server's own bit b_i, it considers 0 as the result if $b_i = 0$ and keeps $E(b'_i)$ as the result if $b_i = 1$. Here the corresponding group encryption of $E(b'_i)$ is performed on the bit b_i. The results after the '$\oplus$' or '$\cdot$' operation, as of now, may reveal information about the underlying ciphertexts used. To secure this, the server then blinds the result adding encryption of a random value $E(v)$ from the corresponding subgroup. $v \xleftarrow{R} G_1$ and E_g encryption is used if $E(b'_i) \in G$ else $v \xleftarrow{R} H_1$ and E_h encryption is used if $E(b'_i) \in H$.

The multiplication operation in *Step 6* is performed between the encrypted bits of $m'_i \cdot m_i$ and $b'_i \oplus b_i$ which belong to G and H respectively. The server then adds up the corresponding encrypted bits to obtain $E_t(d_{Hn})\ and\ E_g(d_{Hd})$ - encrypted numerator and denominator of the normalized hamming distance.

The threshold(τ) is represented as a fraction(τ_n/τ_d) as only integer operations are supported by the cryptosystem. In *Step 7*, the additive homomorphism allows computation of $E_t(\tau_d \times d_{Hn})$ and $E_g(\tau_n \times d_{Hd})$. Next, the homomorphic multiply with $E_h(1)$ converts $E_g(\tau_n \times d_{Hd})$ to a G_t group

encryption. Then the server computes the value $E_t(\tau_d \times d_{Hn} - \tau_n \times d_{Hd})$ using the additive homomorphic properties of E_t. Now, to decide the authentication result it needs to know if the encrypted value is *positive* or *negative*. The server adds 2048 to the encrypted value to keep it in a non-negative domain helping the decryption, as described in *Section 4.2*. To hide the value from the client the server also randomizes it.

3.6 Protocol analysis

In this section we perform a detailed analysis of the proposed protocol. We discuss the correctness of the secure protocol to be same as a plaintext matching protocol. We analyze the computational complexity of the protocol and explain real time computability with it. We also analyze the security aspects through various attack scenarios during the working of the protocol.

3.6.1 Correctness

The encryptions and randomizations performed during the protocol can be safely decrypted. The cryptosystem provides projection functions - $\{\pi_1, \pi_2, \pi_t\}$, also called trapdoors, when applied on the ciphertexts eliminate the random values added while performing mathematical operations. Computing a discrete logarithm of those with the base of the logarithm set to the projection values(π) of the corresponding group generators successfully outputs the encrypted message.

The correctness of the protocol also depends on the following binary bit operations that hold true:

$(b \oplus b') \oplus b = b' \quad ; \quad a \cdot (b \oplus b') = (a \cdot b) \oplus (a \cdot b')$

They support the template protection scheme used and the randomization of the mask vector at *Step 3*.

3.6.2 Computational analysis

The encryption-decryption, homomorphic additive and multiplicative operations are the expensive ones among the total computations performed during the complete execution of the protocol. Optimizing them in any way can reduce the live authentication time of the protocol. We suggest offline computations and optimizations in order to reduce the online time taken by the system. The encryption operations can be pushed offline. The client can pre-compute sufficient number of encryptions of $0's$ and $1's$ before the start of the process. This reduces the encryption time in the online setting to $\mathcal{O}(1)$. The homomorphic additions require negligible time compared to the bilinear pairings in the multiplication operation. The pairings can't be pushed offline or pre-computed as both its inputs are ciphertexts, available only after *Step 6* of the protocol. The number of pairings to be computed can be reduced.

As defined in *Section 2.1*, 3 pairings are computed per multiplication operation where $G_g \times G_h \rightarrow G_t$ action is performed and 2 pairings are computed per addition for a ciphertext $C \in G_t$. The actual multiplication operation requires one pairing , the rest are for randomizing the output. Similarly, the

actual addition operation in G_t does not require any pairing. The server computes these operations in *Step 6* of the protocol. We reduce the computation to just one pairing per operation and, in the end, perform a single randomization of the ciphertext.

The decryption operation requires computing discrete logarithms. Computing discrete logarithm of any integer has a complexity of $\mathcal{O}(\sqrt{s})$ using the Pollard's *rho* algorithm; where s is the search space size. The client needs to compute the discrete logarithms of the ciphertexts received in *Step 3* and *Step 7* of the protocol. We propose the client to perform pre-computations of the exponents using its private key to reduce the complexity to $\mathcal{O}(1)$. The number of pre-computations required depend on the range of p and r chosen by the server. We propose to limit the values to $\mathcal{O}(10^3)$ causing the value '$\alpha \times p + r$' to be $\mathcal{O}(4.097 \times 10^6)$. Keeping it in a non-negative and definite range allows successful decryption using the pre-computations. Storing pre-computed values by hashing allows quick decryption of the randomized distances.

3.6.3 Security and privacy analysis

Biometric systems are known to be more secure compared to password based systems as biometric data is difficult to reproduce. The proposed system adds a token requirement for authentication directly enhancing the security. The underlying security of a biometric system depends on its design and the security of the underlying cryptographic schemes used. Its design may provide various points of attack for an adversary without valid credentials. The privacy offered depends on the data revealed while undergoing the authentication protocol. The protocol that we describe is secure against a semi-honest adversary. A client which deviates from the prescribed steps can't learn any information about the biometric template at the server. We show that the chances of such deviations leading to acceptance are minimal.

- ***Privacy:***

  The privacy offered by the protocol depends on the information revealed to the server about the client carrying out the authentication and the information revealed to the client about the user templates present at the server. During the execution of the protocol the clients biometric data is kept encrypted all the time. The values decrypted by the client in *Step 3* are randomized by the server, thus not revealing any information about the server template. Also, in *Step 8* the randomized matching score is received by the client, hiding the actual matching result. Thus, complete privacy of the server template is offered by the protocol. Also, complete privacy of the client biometric and the keys is offered except that the matching score of the information provided by the client with the stored template is revealed to the server.

- ***Server security:***

  Case 1: Adversary gains access to the template database - The biometric templates present in the server database are all XORed with the keys generated by the enrollment server. The keys used are

randomly generated making the biometric information stored completely random. Thus, they do not reveal any information about the user's biometric. A user may be registered at different servers which use this same protocol. The templates present there will be computed using different keys. Even if the adversary manages to break into multiple databases, no information about the user's biometric is revealed. Whenever a template is suspected to be broken, a new one with a different key can be generated.

Case 2: Adversary is in the server during the authentication - During an authentication process, the data received from the client is encrypted, keeping it secure. The adversary will get to know the random values generated by the server, also the final 'β'. These intermediate values will not reveal any information about the client's biometric B' or M'. However, the final score revealed will not assist the adversary for carrying out further attacks as it has no information about the vectors used to calculate the score.

Case 3: Adversary carrying out brute force attacks - The complexity of the brute force attack will be of the product of the plain biometric and the keys. However, in the final step the $'\beta'$ can sent to pass the authentication test. The probability of successfully passing the server divisibility test by sending a suitable β is *one* in $\mathcal{O}(10^6)$ for each every authentication which is small.

- ***Client-End security:***

Case 4: Adversary gains access to user's biometric or authentication keys - An adversary having access to the client's system cannot carry out authentication without access to both the biometric and the key. If adversary has access to one, the amount of effort required is equal to randomly guessing the other bit vector.

Case 5: Client trying to gain information about the server template If the client modifies the bits to be sent to the server hoping to learn some information of the biometric present at the server, the randomization in *Step 2* will blind the server vectors from the client. In *Step 8* of the protocol, the client learns the randomized value '$\alpha \times p + r$'. He can send a modified value to pass the authentication test. However, the probability of successfully modifying it to pass the server divisibility test is small.

- ***Network security:***

Case 6: Adversary gains access to the network - An adversary sniffing over the network can only access the encrypted biometric data or data computed using the random numbers generated by the server. This will not help in deciphering any information about the biometric templates. The acquired data will not be useful to host a replay attack as the client's supplied data need to be in sync with the server's random keys for the session.

Curve	*Security*	*Client*	*Server*			*Bandwidth*
			Pairing	*Rest*	*Total*	
MNT	80 $bits$	10 ms	57.1 s	0.9 s	58 s	400 KB
BN	128 $bits$	10 ms	90.6 s	1.4 s	92 s	640 KB

Table 3.1 Security, computation and bandwidth requirement with different pairing schemes.

3.7 Experiments

3.7.1 Implementation and results

For evaluation purpose, the protocol was implemented on a *client-server* architecture in C++ using the MIRACL library. The experiments were performed on a single core of an Intel $3.2Ghz$ processor with around 200 KB of memory usage. The public ICE 2005 database [5] was chosen which has 2953 iris images from 243 eyes. For palmprints, the PolyU database consisting of 7752 images from 386 people was chosen. The homomorphic algorithm was implemented using the type-3 pairing API provided by the library. The pairing based groups were generated for two variants of curves: 1. MNT curve which provides security equivalent to an 80 bit AES encryption and, 2. BN curve which provides security equivalent to an 128 bit AES encryption. The iriscodes and palmcodes, extracted using the algorithms described in *section 2.2*, both consist of a 2048 bit binary feature vector and a same size mask vector. To verify the template matching performance of our proposed system, we compared our results by performing the matching on plain text iriscodes and palmcodes. Both produced the same GAR and FRR values.

All pre-computations and encryptions mentioned in *Section 4.2* were implemented offline. The results are mentioned in *Table 3.1*. Iriscodes and palmcodes, both are same size vectors so their matching times are same. The server values are split into time for bilinear pairings in the multiplications and time for rest of the operations. These server computations are inherently parallelizable and one can achieve near-linear speedup w.r.t. number of cores.

We compare our results with a recent efficient implementation of a secure biometric matching system, which supports template protection in *Table 3.2*. For the scheme in [79], the values are generated on their 1024 bit public encryption, feature vector size of 2048 $bits$. We don't take into account their dedicated hardware accelerations. We compare their results with our 80 bit AES security implementation. To the best of our knowledge, there is no other work which supports template security and also involves a mask vector, providing specific details of computation and communication costs.

3.7.2 Alternate method

Garbled circuits allow construction of an alternate solution to the addressed problem. [40] provide efficient implementation of the same. The final score, which is revealed to the server in the current

Scheme	*Time*	*Bandwidth*	*Comm. Rounds*
[79]	$4\ min$	$5\ MB$	2
Proposed	$58\ sec$	$400\ KB$	3

Table 3.2 Comparison with existing similar systems.

execution, can be masked from the server with their help. The main drawback using garbled circuits is the trust issue. In the two party setting, the party evaluating the circuit with its own input has to trust that the circuit provided performs the expected task. Acquiring the biometric data or template supplied by the other person by providing him with a fabricated circuit is easy. The solution to this is using verifiable garbled circuits. Construction of such circuits is complex and computationally inefficient.

3.8 Summary

We propose a novel protocol which is able to achieve secure and private authentication and also provides template protection without loss of accuracy of the matching algorithm. If offers template revocability, if one is suspected to be broken. The user provides his identity only at the time of enrollment. Once the trusted server secures the templates, no information about the user biometric is leaked. We measure the efficiency of the protocol by implementing it on a *client-server* architecture and estimate the computation and communication complexities. We allow linear scaling of the protocol by increasing the number of processors. This permits verification to be performed in real time with the help of available hardware and proposed pre-computations.

Chapter 4

One-Time Biometric Token based Authentication

4.1 Introduction and related work

The idea of one-time passwords(OTPs) emerged to improve traditional password based authentication, where an individual's password leakage directly compromises his system's security. The person can still change his password, but till then it could be too late. If financial accounts are involved, the loss of password for even a small period of time can cause huge losses to the individual. Often used techniques of smart-cards, pins and OTPs sent to the cellphone can be hijacked and individuals can be impersonated with a simple theft of device. We have discussed that biometrics based systems, on the other hand, provide a reliable solution for recognizing individuals. With the property of uniqueness that biometrics provide, the risk of permanently losing one's biometric trait through adversarial attacks has been discussed in the earlier chapters. In this chapter we propose secure one-time biometric tokens that bind the two systems and can be safely used as OTPs. We describe the protocol in the framework of an online bank transaction carried out by an individual at some service provider.

One-time passwords based on time stamps, discussed by Haller *et al.* [37] have been extended to biometrics to develop the concept of one-time biometric templates. To the best of our knowledge, they were introduced by Ueshige *et al.* [76], where they require a one-time-transform generating server. It communicates a common transaction based one-time-transform function to both the authentication server and client and the protocol happens over several rounds of communication. They suggest invertible transforms to be applied so as to preserve the accuracy of the biometric match. Also, their one-time-transforms require an earlier biometric feature value. Any specific implementations have not been provided in their work.

One-time templates for face features have been provided by Lee *et al.* [51]. They propose repeatable transforms based on orthogonalizing and randomizing biometric features. Each authentication generates a new transform based on an user specific secret. The subsequent authentication attempts must update current transformation vectors present at the client and the server for a successful authentication. The matching accuracy is upheld even while authenticating in the transformed domain.

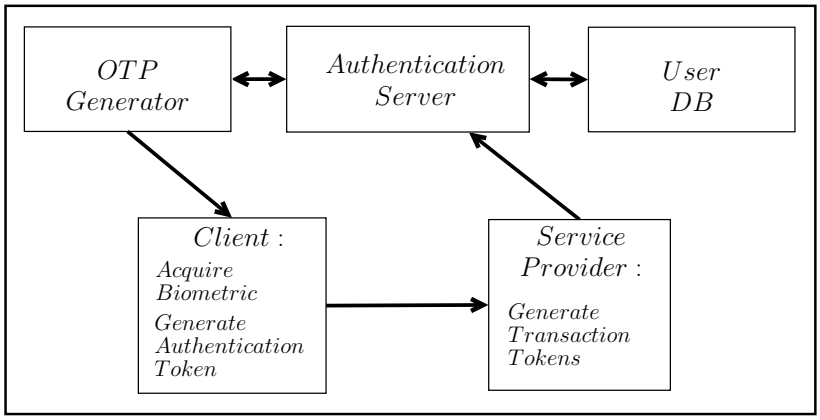

Figure 4.1 General One-Time Password Authentication

Bringer *et al.* [16] propose anonymous time based authentication with cancelable biometrics. Their work does not rely on user specific keys for transforms, rather they assume the scanning hardware to be capable of generating one-time credentials. On every authentication attempt the scanner must communicate with the authentication server to compute time specific cancelable transformations. They propose certain properties of the distortions which prevent using one sensor's transforms onto another sensor, however implementation details of such transform functions are not provided.

Our proposed protocol is based on error correction codes, also recently used in [56, 48]. However current implementations cannot be directly used for OTP based transaction authentication. In our proposed authentication scenario, the server holding the biometric credentials, on receiving a request, provides a random one-time key to the client. The client then computes an authentication token based on the received key and its biometric information, the one-time biometric tokens are then transferred to the service provider to execute the transaction. In *Section 4.2*, we discuss the structure of one-time password scheme. In *Section 4.3*, we discuss the error correction scheme used. We describe our protocol in *Section 4.4* in the scenario of a bank transaction. In *Section 4.5*, we perform experiments on the iris biometric trait to evaluate the performance of our protocol. In *Section 4.6*, we perform a detailed correctness and security analysis of the protocol and then summarize our work in *Section 4.7*.

4.2 One-time passwords

We describe a general one-time password based transaction scheme in *Figure 4.1*. It comprises the following entities - *(i)* OTP Generator, *(ii)* Authentication Server, *(iii)* User Database, *(iv)* Service Provider and *(v)* Client interface. The storage module and the one-time password generator can be together with the authentication module or exist as separate entities depending on the protocol implementation. When biometric authentication is included in process, involved biometric data must be secured against adversarial attacks of eavesdropping, substitution and impersonation. One or multiple communication links can be attacked, hence the one-time passwords, biometric tokens and transaction

data must be implicitly secure, without additional requirement of transport layer security(TLS) to ensure that no biometric information is leaked. Public certificates are generally issued only to the servers due to difficulties in their validation and clients' authentication is based only on their biometrics and tokens. During the authentication process, the server side, on receiving a request, provides a one-time key to the client. It is possible to serve this one-time password request without any biometric verification as the crucial part is the validation of final transaction, also it reduces the computation overhead of the OTP generator. Generating completely random OTPs, independent of biometric templates also reduces computation and communication overhead and avoids biometric information leakage when they are being transferred to the client. Next, the client's biometric information and bank credentials need to be secured before providing them to the service provider. To uphold the privacy of client's bank credentials, transaction anonymity needs to be provided with respect to the service provider. Even during the final authentication at the server, details of the biometric data should not be revealed. We develop our protocol considering these aspects of one-time password based authentication.

4.3 Error correction

The feature vectors extracted from different biometric instances of the same individual certainly have some differences. We attempt to correct the differences of the query instance to exactly match to the stored template through error correction. To implement the process we use a two layered error correction scheme. The first or the top layer handles random errors throughout the code and the second or the inner layer handles burst errors. This scheme is similar to the one proposed in [38] where they use Hadamard codes to correct random errors and Reed-Solomon codes to handle error bursts (localized errors). A pseudo random key of length $(K * b)\ bits$ is generated and encoded using Reed Solomon encoding to output a $(N * b)\ bit$ Reed-Solomon code. The obtained values are further encoded using the Hadamard linear error correction code, which is repeatedly applied over block length b generating $(2^{b-1})\ bit$ codes for each block. The final obtained code of $(N * 2^{b-1})\ bits$ is then XORed with the biometric vector to obtain the secure code.

The Reed-Solomon encoding is based on the parameters b, K and N and satisfies the following condition:

$$K = N - 2T \tag{4.1}$$

Where, the input message consists of K blocks, the output code has N blocks, the error handling capacity of the code is T blocks and the block size is $b\ bits$.

The Hadamard encoding outputs a $(2^{b-1})\ bit$ code for a $b\ bit$ message with the correction capacity of $(2^{b-2} - 1)\ bits$.

While decoding, a modified code is received by the decoder. The obtained code is split into the blocks of $(2^{b-1})\ bits$, decoded by finding the closest linear code using the Hadamard matrix which maps each block to a $b\ bit$ code. The linear decoder performs most of the error correction. There exist several burst errors, where the complete linear coded block may be erroneous. They cannot be corrected

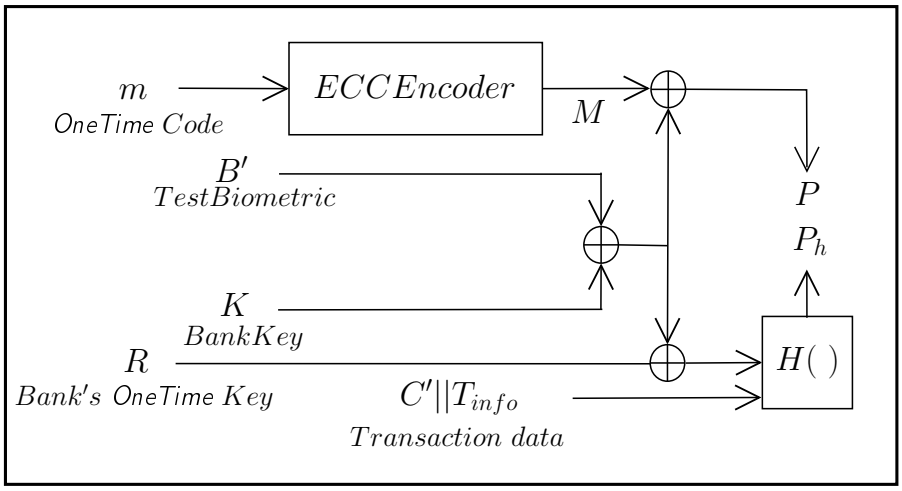

Figure 4.2 One-Time Biometric Token

using the linear correction, the RS decoding in the second part handles them. If the localized errors are within the capacity of the RS code, the decoding will output the perfect original key. Large number of errors will output a different unrelated key.

4.4 Authentication protocol

The protocol describes the steps to compute a one-time biometric token and use it to carry out a bank transaction with a service provider. The user is authenticated by the bank based on his biometric trait and an authentication key present in a smart-card issued to him, which also has his bank identity. The service provider is authenticated using its identity and an issued key. We expect the bank and the service provider to have public certificates signed by a trusted authority, as this has become a standard for all internet transactions today. As anywhere else, these certificates are used by our protocol to verify the authenticity of the party one is communicating with. They are not used to encrypt the one-time keys or biometric data of the user, however these are used by the protocol to bind the user's one-time transaction token with the specific service provider for enhanced security.

In our workflow, the user first verifies the service provider and his bank through their certificates. He then requests the bank to issue him a transaction ID and a one-time key. He then scans his biometric information, uses the smart-card key and a randomly generated code to create an authentication token. The bank's received one-time key is binded using the biometric, the smart-card key, the service providers certificate and the transaction information using a cryptographic hash. This generated pair of values along with the transaction ID is the one-time authentication token(*Figure 4.2*) of the client which is sent to the service provider. The service provider sends it to the bank along with his authentication credentials and the transaction information. The bank then verifies the validity of the transaction request to complete the authentication process. The protocol is described below in detail.

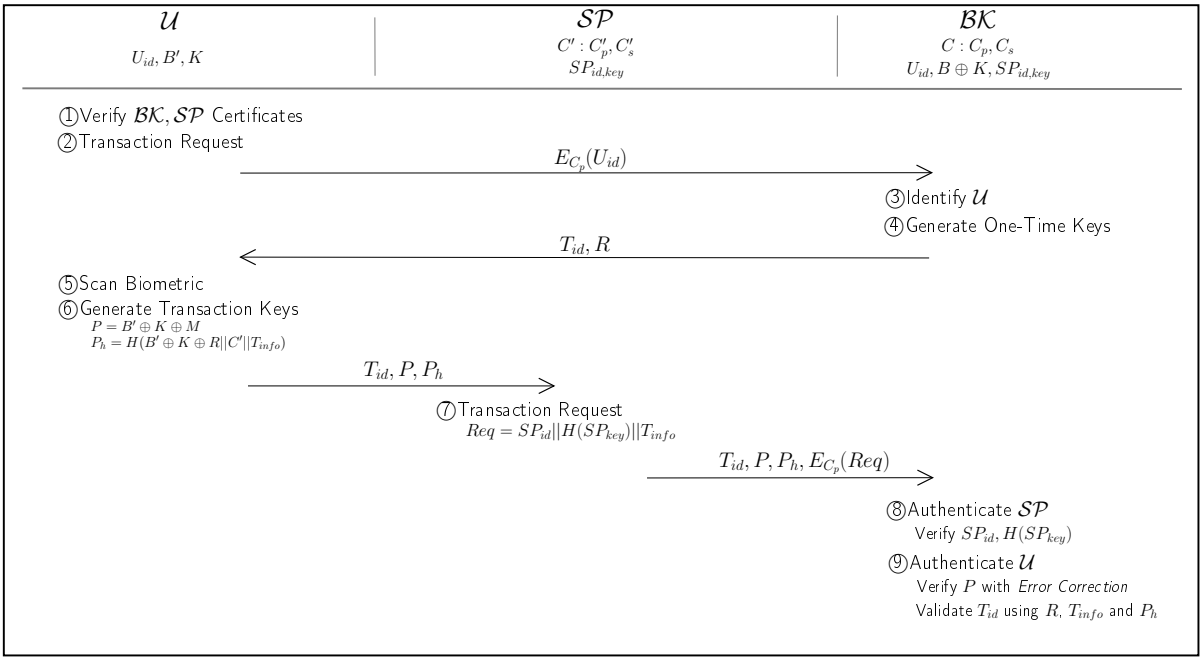

Figure 4.3 Proposed one-time biometric token based authentication protocol.

4.4.1 Registration phase

The registration process takes place in a secure environment. A user $\mathcal{U}$ registers with the bank $\mathcal{BK}$ by providing its biometric B. The bank creates the user's identity U_{id} and token K and stores its biometric template as $B \oplus K$. The user's identity and token are communicated to the user through a smart-card. The service provider registers at the bank to acquire its identity SP_{id} and password SP_{key}.

The bank's public certificate C contains its public key C_p and secret key C_s. Similarly, the service provider's certificate C' has its public key C'_p and secret key C'_s.

4.4.2 Authentication phase

The client initiates the authentication phase when it wishes to generate a one-time password for a financial transaction. The steps carried out in the authentication are described below.

INPUT:
$\mathcal{U}$: U_{id}, Biometric B', Token K.
$\mathcal{BK}$: Biometric template $B \oplus K$, Certificate C - Public Key C_p, Secret Key C_s.
$\mathcal{SP}$: SP_{id}, SP_{key}, Certificate C' - Public Key C'_p, Secret Key C'_s.
OUTPUT:
$\mathcal{BK}$ returning a *Success* or *Failure* after validating the transaction between $\mathcal{U}$ and $\mathcal{SP}$.
PROTOCOL:

1. $\mathcal{U}$ first verifies the $\mathcal{SP}$ and $\mathcal{BK}$ certificates C' and C to be signed by a trusted signing authority. It then requests $\mathcal{BK}$ for a one time key by sending his U_{id} encrypted with C_p.

2. $\mathcal{BK}$ identifies the U_{id} decrypting the request. It generates a transaction ID T_{id} and a random binary R of length equal to the stored template and maps them with the received U_{id}. The unique T_{id} and R are sent to $\mathcal{U}$ as the transaction key.

3. $\mathcal{U}$ receives T_{id} and R. It generates a random key m and computes an error correcting code M as per the scheme mentioned above. It extracts a feature vector B' from his biometric instance and XORs it with its token K along with M to get $P = B' \oplus K \oplus M$. P is the user's password for authentication. To generate the per transaction token, it computes a vector $B' \oplus K \oplus R$. It also concatenates C' and T_{info} to it to bind this transaction to that particular SP. T_{info} contains the details of the transaction being performed. It then computes $P_h = H(B' \oplus K \oplus R||C'||T_{info})$ which acts as a verification hash for the transaction. It sends T_{id}, P and P_h to $\mathcal{SP}$.

4. $\mathcal{SP}$ receives T_{id}, P and P_h and sends the signed hash of the received value as a receipt. It verifies $\mathcal{BK}'s$ certificate and creates a request for a transaction, $Req = SP_{id}||H(SP_{key})||T_{info}$. Encrypted request $E_{C_p}(Req)$ is sent to $\mathcal{BK}$ along with the T_{id}, P, P_h received from the user.

5. $\mathcal{BK}$ receives the request, identifies $\mathcal{SP}$. Authenticates it by matching stored SP_{id} and SP_{key} with the received ones. Next, it verifies the user data. Using the received T_{id}, $\mathcal{BK}$ obtains the corresponding U_{id}, the one-time key R and the biometric template $B \oplus K$ corresponding to the U_{id}. $\mathcal{BK}$ performs a XOR, $P \oplus (B \oplus K)$ i.e. $B' \oplus K \oplus M \oplus B \oplus K$, to obtain a modified error correcting code M'. It decodes M' using the decoding method mentioned earlier to get the key m'. If this decoding is unsuccessful $\mathcal{BK}$ aborts the transaction. A successful decoding does not mean a successful authentication as the decoding may have lead to extraction of a m' different from the users m due to the variations in B'. $\mathcal{BK}$ encodes the received m' again to generate a M''. It performs $M' \oplus M''$ to obtain the hamming distance vector D. This distance is the difference between the two biometric samples. It then verifies the hamming distance to be less than the matching threshold. If this fails $\mathcal{BK}$ aborts the transaction. For a successful authentication, the template, one-time R and P_h are verified as follows. The template $B \oplus K$ is XORed with R and the distance vector D to obtain $B_v = B \oplus K \oplus R \oplus D \sim B' \oplus K \oplus R$. The $SP's$ certificate and received T_{info}, along with B_v are matched with P_h.
   $P_h \stackrel{?}{=} H(B_v||C||T_{info})$, if true the authentication is a *success* else a *failure*.

4.5 Experiments

The proposed protocol can be executed on top of any biometric trait based on binary feature vectors using hamming distance as the distance measure and compatible error correcting codes. We evaluate its matching performance on iris trait by extracting binary feature vectors from a subset of CASIA V3 iris dataset [2] consisting of 100 individuals with 8 samples each. Iriscodes are extracted based on the scheme presented by Ko *et al.* [49] and then binarized using a similar approach as Rathgeb *et al.* 's [65] to output a 2000 bit feature vector. The iris region is first segmented as described by Daugman *et al.* [24]. The segmented iris image is transformed to polar coordinates to obtain a 64×600 image. To remove the occlusion caused due to eye-lids and eye-lashes, only the part in the range $[135^{\circ}\ to\ 225^{\circ}]$ and $[315^{\circ}\ to\ 45^{\circ}]$ is selected as the region of interest. Of the cropped 64×300 image the central 60×250 pixels are considered for feature extraction. The 60×250 image is divided into rectangular cell regions, each containing 3×10 pixels. The average value of the pixels in each cell are used to represent the cell. A cumulative sum based change analysis as discussed in [49] is performed on the obtained 20×30 representative matrix to obtain a feature representation of length 1000. The values of the feature vector $\in \{0, 1, 2\}$. We then binarize each value into two bits, $0 \rightarrow 00,\ 1 \rightarrow 01$ and $2 \rightarrow 10$. The binary vector is then rearranged in the form of two halves, first half consisting of the first bits of the 2 bit binary value and the second half consisting of the second bit. This places together the '1' $bits$ in the binary feature. Such a grouping helps in error correction in our two-layer error correcting code. While computing a XOR operation, the burst errors occurred will result in lesser faulty blocks when rearranged, as compared to the initial binary representation. This will help the block level decoding function to maintain lower correction constraints in turn allowing for longer keys. Rathgeb *et al.* [65] have used a single RS-block encoding for error correction but we were unable to achieve high matching accuracy with that.

In the enrollment stage three iris images are preprocessed to extract corresponding feature vectors and a majority voted bit values are considered to create the template vector for each individual. The 2000 bit binary vector is extended to a 2048 bit vector by inserting 0 $bits$ as we use a 2048 bit error correcting code. The templates are then XORed with individual specific keys of equal length. During authentication, only one iris sample is captured, preprocessed and binarized as discussed above to obtain the query vector. It is XORed with the individual's key and the one-time error correcting code. The decoding is verified as discussed in our protocol in *Section 4.4*. The matching accuracy of the system is evaluated at different key lengths of the error correcting code. The genuine comparisons are used to recover the encoding keys to confirm the match. The key length of the code is varied using both the RS code parameters and the Hadamard matrix size. The results obtained are mentioned in *Table 4.1* and *Table 4.2* when evaluated on a systems with CPU speed of 2.1 GHz and 2 GB RAM. As the key size is increased, the error correction capacity of the code reduces, increasing the FRR of the system. Increased template entropy due to user specific keys prevents any False Accepts.

Key Length ($bits$)	*m* ($bits$)	*FAR* (%)	*FRR* (%)
112	8	0.00	4.78
96	8	0.00	0.87
140	7	0.00	6.09
126	7	0.00	3.04
112	7	0.00	2.61
98	7	0.00	1.30

Table 4.1 Matching performance of the proposed protocol on CASIA V3 Iris evaluation subset at different key lengths. (b - RS code block size)

	Time (in *ms*)
Client	8
Service Provider	4
Server	70

Table 4.2 Computation costs of the proposed protocol.

4.6 Protocol analysis

The computation costs of the protocol are described in 4.2. The protocol does not involve any homommorphic operations or secure two-party computations, thus keeping the computation costs low. We analyze in detail the privacy and security offered by the protocol along with its correctness below.

4.6.1 Correctness

In the execution of the protocol, tasks of validating user's biometric one-time token and service provider's credentials are performed at the bank. The verification of the service provider is a straight forward matching of keys. The credentials of the client are the key encoded biometric feature P, which validates biometric sample, and the hash value P_h, which validates the one-time-ness of the authentication. The decoding capacity of the code is comparable to the system threshold. If a query sample provided by some user consists of errors greater than the correcting capacity, it results into a decoding failure or a generation of an incorrect key. Re-encoding the generated incorrect key with the same error correction scheme will certainly not output a code which matches with the earlier obtained code within the limits of the threshold. So, if a valid decoding takes place, the hash P_h will be exactly matched in the verification. Thus a query which would have been rejected by the biometric system without template protection will surely be rejected by our system.

4.6.2 Privacy analysis

The protocol ensures user anonymity from the service provider. The user does not provide any of his bank credentials to the service provider directly, only binded with the one-time keys. Consolidating the data over several users' transactions, the service provider can XOR the tokens to obtain differences between them. Identifying tokens which are from same individual from the rest is not feasible as the keys used are strong(*Table* 4.1).

4.6.3 Attacks and security analysis

A *dictionary attack* is among the easiest of the attacks attempted, where an adversary can try out some database of iris images against the stored templates to obtain illegitimate authentication. In our protocol the templates are XORed with user specific keys that defend such an attack. XORing with a key raises the difficulty of an dictionary attack to a brute force attack of the order of template size. The key used, in a way, randomizes the underlying biometric template and increases the inter class entropy while maintaining intra class variations.

- ***Client security:***

  The protocol ensures biometric security from an illegitimate user by ensuring that the client does not receive any biometric information during the authentication process. One-time transaction ID and random one-time key are communicated to it, only providing additional security from replay attacks.

- ***Server security:***

  The biometric templates stored at the server are XORed with user specific keys hiding the underlying biometric feature vector. Breach of the server database will only reveal the secured biometric templates to the adversary. For different banks using the same authentication system, the user keys will be different. So a breach of multiple databases will not enhance the chances of leaking of original biometric information. If an adversary is present at the server during the authentication process, he has access to both the template and the query vectors. The intermediate information revealed consists of the original error correcting code used by the client, the distance vector and the matching score. The distance vector just reveals the indices where the query and probe samples matched or differed. Thus the adversary cannot obtain the original bits in the biometric feature vector using any of the intermediate values.

- ***Network security:***

  An adversary may try to break in the network and sniff the data transferred or try and impersonate the bank or the service provider to get hold of user credentials. Impersonation attacks are defended at the step of certificate validation. Certificates issued only by a standard authority are trusted. Transaction information that can be picked up from the network are the random key sent by

the bank(R) and the user's authentication token(P, P_h). The service provider's credentials and transaction data is sent to the bank through a public key encryption. Brute force attacks on P are not feasible due to strong encoding keys involved. If multiple authentication instances can be collected by an adversary, they would be encoded using different encoding keys, resulting into a different error correction code. If we consider the underlying biometric template to be exactly the same, on XORing the two authentication instances will reveal the locations at which the two codes differ. This can be extended to predict the locations at which the encoding keys differ, however no bit of the underlying key is revealed. Intra class variations in the biometric samples provide sufficient entropy to mask direct leakage of such information.

In the worst case, the adversary steals an individual's smart-card key, the transaction tokens and one-time keys sent by the bank. Then the scenario shifts to breaking a key-binding scheme involving the iris feature vector. The security of the key-binded iriscode depends directly on the length of the encoding key, but the iriscode involved is not equivalent to a completely random vector. As discussed in the implementation section, it consists of 200 groups with 5 values per group and the values $\in \{0, 1, 2\}$. The $1's$ and $2's$ are continuous inside the group and the rest of the values are 0. These consecutive $1's$ or $2's$ can start and end at any index from 1 to 5. Thus the allowed combination of values per group is 20. In general, a completely random $5\ bit$ key provides security equivalent to $5\ bits$ as all permutations(32) of $0's$ and $1'sare$ permitted. Then, in our case, the effective security of the 5 values of the iriscode provide security equivalent to $3.12\ bits$. The 5 values, when binarized, contribute $10\ bits$ to the iriscode. Thus, $10\ bits$ of an iriscode provide effective security equivalent to at least a $3\ bit$ random key. Considering that the iriscode is XORed with an error correcting code, we'll have to consider its error correction capacity to effectively compute the security provided by the iriscode. Let the feature vector be $N\ bits$ and the error correction capacity of the code be $\theta\ bits$. Then, for an adversary, to successfully decode the key-binding it has to exactly predict the $(N - \theta)\ bits$ of the iriscode. From the values discussed above, a $(N - \theta)\ bit$ iriscode will effectively provide security equivalent to a $3 * (N - \theta)/10\ bits$. The error correction capacity of the code in turn depends directly on the encoding key length. Shorter the key involved, larger number of corrections are possible for a fixed code length thus lower the security. Considering the key-lengths provided in *Table 4.1*, a $96\ bit$ key will output a code with a correction capacity of $690\ bits$. Then the effective security provided by the $2000\ bit$ iriscode is $493\ bits$ which is equivalent of saying that the adversary has to exactly predict a $493\ bit$ string to break the key-binding. Considering the computation required for that task, a brute force attack is unfeasible.

4.7 Summary

In this chapter, we have proposed a one-time biometric token based authentication scheme. This protocol has been detailed in a scenario of a online bank transaction, where the user generates a one-

time biometric token based on a random one-time key received from the bank. It provides a strong alternative to the currently used methods where user's sole bank credentials or card details need to be provided to service providers. Proposed way of transaction upholds user's biometric privacy and provides anonymity while dealing with any service provider. The bank does not transfer any biometric related information to the user during the execution of the protocol, it follows currently used methods of just sending a random key and mapping it with the user and the transaction ID. Unlike previous protocols it does not require updating biometric templates or keys after each transaction and works within the communication framework of current online banking systems. Performance and security of the system has been analyzed and it meets desired standards.

Chapter 5

Conclusion and future work

All the research work around biometrics has established that several biometric traits play a pivotal role in the task of human identification. The permanent nature of biometrics thus raises the questions of privacy and security when deploying biometric based authentication systems. We have discussed these concerns in the 1^{st} chapter. The major developments to address these concerns and perform reliable authentication have been discussed in the 2^{nd} chapter. Our work in this thesis attempts to contribute to it by providing secure and privacy preserving authentication schemes with protected biometric templates. We have focused on fixed-length binary feature representations as these protocols can then be applied to any biometric trait that can be reliably represented using such feature vectors.

In the 3^{rd} chapter, we propose a two-party authentication protocol based on homomorphic encryption for binary template representations involving masks. The limitations of existing schemes to provide secure, privacy preserving, real-time authentication with template protection have been discussed. Implementation and experimentation of the proposed protocol with iris and palmprint traits has been evaluated to prove our claim. The detailed computation, security and privacy analysis shows that the proposed scheme achieves desired standards. One future direction is to extend this work to $1 : N$ identification. Providing efficient protocols for identifying the individual's query by searching through the database and maintaining the security and privacy standards. Currently, the encryptions and homomorphic operations are computed on each bit individually which causes high computation bandwidth costs. Possibilities to pack the bits and then perform cryptographic operations on blocks can be explored for more efficient protocols.

In the 4^{th} chapter, we propose biometrics based one-time authentication tokens that suit the online bank transaction scenario. Individuals' biometric credentials are securely enrolled at the bank. The protocol discusses the bank transaction of an user with a service provider, where the user creates the secure one-time biometric token using a random one-time key received from the bank and binds it with the transaction information. The detailed security analysis results uphold the security and privacy of biometric features during the authentication process. The experiments conducted with iris trait provide acceptable authentication results of the system. More stable feature representations with lesser intra-class variations will help to achieve higher performance by reducing the false non-match rates.

Related Publications

- Rohan Kulkarni and Anoop M Namboodiri, "Secure Hamming Distance based Biometric Authentication", in Proceedings of the IAPR International Conference on Biometrics (ICB), 2013
- Rohan Kulkarni and Anoop M Namboodiri, "One-Time Biometric Token based Authentication", in Proceedings of the Indian Conference on Computer Vision Image Processing and Graphics (ICVGIP), 2014

Bibliography

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

[2] Casia iris dataset. http://biometrics.idealtest.org/dbDetailForUser.do?id=3.

[3] Unique identification authority of india. http://uidai.gov.in/.

[4] United arab emirates deployment of iris recognition. http://www.cl.cam.ac.uk/ jgd1000/deployments.html.

[5] National institute of standards and technology (nist). In *Iris Challenge Evaluation. http://iris.nist.gov/ICE*, 2005.

[6] J. Anil K et al. Biometric template security. *EURASIP Journal on Advances in Signal Processing*, 2008.

[7] E. Barker et al. Recommendation for key management–part 1: General (revised). *NIST special publication*, 800:57, 2011.

[8] M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. D. Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, A. Piva, et al. A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingercode templates. In *4th IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS)*, pages 1–7. IEEE, 2010.

[9] M. Blanton and P. Gasti. Secure and efficient protocols for iris and fingerprint identification. *Computer Security–ESORICS*, pages 190–209, 2011.

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

[11] A. C. O. Bock. Automatic fingerprint machine, 1925. US Patent 1,529,864.

[12] D. Boneh, E. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. *Theory of Cryptography*, pages 325–341, 2005.

[13] T. Boult. Robust distance measures for face-recognition supporting revocable biometric tokens. In *7th International Conference on Automatic Face and Gesture Recognition*, pages 560–566. IEEE, 2006.

[14] T. E. Boult, W. J. Scheirer, and R. Woodworth. Revocable fingerprint biotokens: Accuracy and security analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

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

[16] J. Bringer, H. Chabanne, and B. Kindarji. Anonymous identification with cancelable biometrics. In *6th International Symposium on Image and Signal Processing and Analysis*, pages 494–499. IEEE, 2009.

[17] J. Bringer, H. Chabanne, and B. Kindarji. Identification with encrypted biometric data. *Security and Communication Networks*, 4(5):548–562, 2011.

[18] J. Bringer et al. An application of the goldwasser-micali cryptosystem to biometric authentication. In *Information Security and Privacy*, pages 96–106. Springer, 2007.

[19] J. Bringer, M. Favre, H. Chabanne, and A. Patey. Faster secure computation for biometric identification using filtering. In *5th IAPR International Conference on Biometrics (ICB)*, pages 257–264. IEEE, 2012.

[20] Y.-J. Chang, W. Zhang, and T. Chen. Biometrics-based cryptographic key generation. In *IEEE International Conference on Multimedia and Expo*, volume 3, pages 2203–2206. IEEE, 2004.

[21] T. C. Clancy, N. Kiyavash, and D. J. Lin. Secure smartcardbased fingerprint authentication. In *Proceedings of the ACM SIGMM workshop on Biometrics methods and applications*, pages 45–52. ACM, 2003.

[22] T. Connie, A. Teoh, M. Goh, and D. Ngo. Palmhashing: a novel approach for cancelable biometrics. *Information Processing Letters*, 93(1):1–5, 2005.

[23] J. Daugman. The importance of being random: statistical principles of iris recognition. *Pattern recognition*, 36(2):279–291, 2003.

[24] J. Daugman. How iris recognition works. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):21–30, 2004.

[25] J. G. Daugman. Biometric personal identification system based on iris analysis, 1994. US Patent 5,291,560.

[26] G. Davida, B. Matt, Y. Frankel, and R. Peralta. On the relation of error correction and cryptography to an off line biometric based identification scheme. In *Workshop on coding and cryptography*, pages 129–138, 1999.

[27] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

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

[29] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft. Privacy-preserving face recognition. In *Privacy Enhancing Technologies*, pages 235–253. Springer, 2009.

[30] D. Evans, Y. Huang, J. Katz, and L. Malka. Efficient privacy-preserving biometric identification. In *17th Proceedings of the Network and Distributed System Security Symposium, NDSS*, 2011.

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

[32] D. Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. *Advances in Cryptology–EUROCRYPT*, pages 44–61, 2010.

[33] W. K. French. Automatic recognition of fingerprints by sensing the skin surface with electrical apparatus, 1966. US Patent 3,231,861.

[34] A. N. G. Palmprint identification system, 1971. US Patent 3,581,282.

[35] F. Galton. Personal identification and description. *Journal of Anthropological Institute of Great Britain and Ireland*, pages 177–191, 1889.

[36] C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

[37] N. Haller, C. Metz, P. Nesser, and M. Straw. A one-time password system. Technical report, RFC 1938, May, 1996.

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

[39] F. Hao, J. Daugman, and P. Zielinski. A fast search algorithm for a large fuzzy database. *IEEE Transactions on Information Forensics and Security*, 3(2):203–212, 2008.

[40] Y. Huang, D. Evans, J. Katz, and L. Malka. Faster secure two-party computation using garbled circuits. In *USENIX Security Symposium*, 2011.

[41] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti. Fingercode: a filterbank for fingerprint representation and matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2. IEEE, 1999.

[42] A. K. Jain, A. Ross, and S. Pankanti. Biometrics: a tool for information security. *Information Forensics and Security, IEEE Transactions on*, 1(2):125–143, 2006.

[43] J. Jeffers and A. Arakala. Minutiae-based structures for a fuzzy vault. In *Biometrics Symposium: Special Session on Research at the Biometric Consortium Conference*, pages 1–6. IEEE, 2006.

[44] S. Jha, L. Kruger, and V. Shmatikov. Towards practical privacy for genomic computation. In *IEEE Symposium on Security and Privacy*, pages 216–230. IEEE, 2008.

[45] A. Jin, D. Ling, and A. Goh. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern recognition*, 37(11):2245–2255, 2004.

[46] A. Juels and M. Sudan. A fuzzy vault scheme. *International Symposium on Information Theory*, 2002.

[47] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *6th ACM conference on Computer and communications security*, pages 28–36. ACM, 1999.

[48] S. G. Kanade, D. Petrovska-Delacretaz, and B. Dorizzi. A novel crypto-biometric scheme for establishing secure communication sessions between two clients. In *Proceedings of International Conference of the Biometrics Special Interest Group*, pages 1–6. IEEE, 2012.

[49] J.-G. Ko, Y.-H. Gil, and J.-H. Yoo. Iris recognition using cumulative sum based change analysis. In *International Symposium on Intelligent Signal Processing and Communications*, pages 275–278. IEEE, 2006.

[50] W. K. Kong, D. Zhang, and W. Li. Palmprint feature extraction using 2-d gabor filters. *Pattern recognition*, 36(10):2339–2347, 2003.

[51] Y. Lee, Y. Lee, Y. Chung, and K. Moon. One-time templates for face authentication. In *International Conference on Convergence Information Technology*, pages 1818–1823. IEEE, 2007.

[52] Y. J. Lee, K. Bae, S. J. Lee, K. R. Park, and J. Kim. Biometric key binding: Fuzzy vault based on iris images. In *Advances in Biometrics*, pages 800–808. Springer, 2007.

[53] Y. Lindell and B. Pinkas. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In *Advances in Cryptology-EUROCRYPT*, pages 52–78. Springer, 2007.

[54] D. Malkhi, N. Nisan, B. Pinkas, Y. Sella, et al. Fairplay-secure two-party computation system. In *USENIX Security Symposium*, pages 287–302. San Diego, CA, USA, 2004.

[55] K. Nandakumar. A fingerprint cryptosystem based on minutiae phase spectrum. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2010.

[56] K. Nandakumar. Biosake: Biometrics-based secure authentication and key exchange. In *Proceedings of International Conference on Biometrics*, pages 1–8. IEEE, 2013.

[57] K. Nandakumar, A. K. Jain, and S. Pankanti. Fingerprint-based fuzzy vault: Implementation and performance. *IEEE Transactions on Information Forensics and Security*, 2(4):744–757, 2007.

[58] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich. Scifi-a system for secure face identification. In *IEEE Symposium on Security and Privacy (SP)*, pages 239–254. IEEE, 2010.

[59] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in CryptologyEUROCRYPT*, pages 223–238. Springer, 1999.

[60] A. Paus, A.-R. Sadeghi, and T. Schneider. Practical secure evaluation of semi-private functions. In *Applied Cryptography and Network Security*, pages 89–106. Springer, 2009.

[61] B. Pinkas, T. Schneider, N. P. Smart, and S. C. Williams. Secure two-party computation is practical. In *Advances in Cryptology–ASIACRYPT*, pages 250–267. Springer, 2009.

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

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

[64] N. K. Ratha, J. H. Connell, and R. M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM systems Journal*, 40(3):614–634, 2001.

[65] C. Rathgeb and A. Uhl. Systematic construction of iris-based fuzzy commitment schemes. In *Advances in Biometrics*, pages 940–949. Springer, 2009.

[66] C. Rathgeb and A. Uhl. Secure iris recognition based on local intensity variations. In *Image Analysis and Recognition*, pages 266–275. Springer, 2010.

[67] C. Rathgeb and A. Uhl. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, pages 1–25, 2011.

[68] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg. Efficient privacy-preserving face recognition. In *Information, Security and Cryptology(ICISC)*, pages 229–244. Springer, 2010.

[69] M. Savvides, B. Vijaya Kumar, and P. Khosla. Cancelable biometric filters for face recognition. In *17th International Conference on Pattern Recognition*, volume 3, pages 922–925. IEEE, 2004.

[70] C. Siew Chin, A. T. Beng Jin, and D. N. Chek Ling. High security iris verification system based on random secret integration. *Computer Vision and Image Understanding*, 102(2):169–177, 2006.

[71] Y. Sutcu, H. T. Sencar, and N. Memon. A secure biometric authentication scheme based on robust hashing. In *Proceedings of the 7th workshop on Multimedia and security*, pages 111–116. ACM, 2005.

[72] A. Teoh, B. Jin, T. Connie, D. Ngo, and C. Ling. Remarks on biohash and its mathematical foundation. *Information Processing Letters*, 100(4):145–150, 2006.

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

[74] A. B. J. Teoh, K.-A. Toh, and W. K. Yip. 2ˆ n discretisation of biophasor in cancellable biometrics. In *Advances in Biometrics*, pages 435–444. Springer, 2007.

[75] V. V. T. Tong, H. Sibert, J. Lecoeur, and M. Girault. Biometric fuzzy extractors made practical: a proposal based on fingercodes. In *Advances in Biometrics*, pages 604–613. Springer, 2007.

[76] Y. Ueshige and K. Sakurai. A proposal of one-time biometric authentication. In *Security and Management*, pages 78–83, 2006.

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

[78] U. Uludag and A. K. Jain. Fuzzy fingerprint vault. In *Proc. Workshop: Biometrics: Challenges Arising from Theory to Practice*, pages 13–16, 2004.

[79] M. Upmanyu et al. Blind authentication: a secure crypto-biometric verification protocol. *IEEE Transactions on Information Forensics and Security*, 5(2):255–268, 2010.

[80] M. Van Der Veen, T. Kevenaar, G.-J. Schrijen, T. H. Akkermans, F. Zuo, et al. Face biometrics with renewable templates. In *Proceedings of SPIE*, volume 6072, page 60720J, 2006.

[81] R. Wildes. Iris recognition: an emerging biometric technology. *Proceedings of the IEEE*, 85(9):1348–1363, 1997.

[82] Y. Wu and B. Qiu. Transforming a pattern identifier into biometric key generators. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 78–82. IEEE, 2010.

[83] D. Zhang, W.-K. Kong, J. You, and M. Wong. Online palmprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1041–1050, 2003.

[84] J. Zuo, N. K. Ratha, and J. H. Connell. Cancelable iris biometric. In *19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.