Chapter 3

Fingerprint Template Protection

3.1 Introduction

Fingerprints are by far the most prevalent characteristics used in biometric authentication systems. Studies involving fingerprints were initiated long back in 1858 [126]. A fingerprint is a pattern of ridges and valleys on the surface of a fingertip, which is formed during the first seven months of fetal development [10]. As illustrated in Figure 3.1, these patterns can be of great variety. The fingerprint patterns of each person are different, including for identical twins. Even fingerprints obtained from different fingers of the same person considerably varies. Another important reason for the widespread use of fingerprint based authentication systems is the low cost of fingerprint sensors. A typical government certified fingerprint sensor costs around Rs. 5000 in India and around $20 in the US. As such, embedding these cheap sensors (compared to sensors suited for other traits) in application devices like laptops

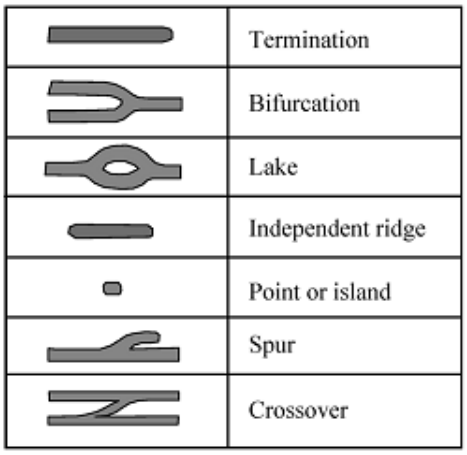


FIGURE 3.1: Discriminating patterns of a fingerprint.

and cell phones is very convenient. Additionally, fingerprints have the remarkable property of being time invariant, which enables the resulting biometric system to be consistent and robust.

The extensive use of fingerprint based biometric systems has resulted in some associated pitfalls. Most importantly, the security and privacy issues of the system users get severely jeopardized. Fingerprint templates are most commonly represented by the extracted minutia points from the fingerprint images. Minutiae features are essentially points representing ridge endings and bifurcations. These points are represented by triplets (x, y, θ), where x represents the abscissa of the point, y represents the ordinate of the point and θ represents the angle which the point makes with the horizontal axis [127]. However, this type of representation of fingerprints is vulnerable to various security attacks including *template inversion*, and *hill climbing*. In a template inversion technique, biometric image features are identified from a stolen template and subsequently used for reconstructing the corresponding biometric image. Alternatively in a hill climbing technique [128], an adversary starts with an

initial guess of the biometric image and then iteratively refines it based on the score obtained by matching the guessed biometric image with the stored one. The severity of these attacks can be gauged by the numerous number of attack techniques developed on these two design principles. A brief summary of such techniques is presented in a tabular form in Table 3.1.

Reference	**Input Representation**	**Output Form**	**Method**
[129]	Minutiae	Fingerprint image	Template inversion
[130]	Minutiae	Fingerprint image	Template inversion
[131]	Minutiae	Fingerprint image	Template inversion
[132]	Minutiae	Fingerprint image	Template inversion
[133]	Matching System	Minutiae	Hill climbing
[134]	Matching System	Minutiae	Hill climbing

TABLE 3.1: Techniques for recovering fingerprint image from stored minutia points.

A host of successful attacks on the stored minutiae points motivated researchers for constructing some transformation based techniques which can encode the original points into some other form. As discussed in Chapter 2, *cancelable biometrics* based schemes solve this problem by designing non-invertible transformation functions. Apart from achieving the required objectives of *non-invertibility* and *unlinkability*, cancelable biometrics based models have the additional benefit of cancelability. This important feature refers to the property that a new template can be generated from the original ones in case the existing template gets compromised in any way. This chapter presents one such scheme for fingerprints which implements cryptographic hash functions as the transformation functions. The proposed framework also consists of other indispensable modules such as 'pre-alignment' and 'quantization' which cater to inherent issues like translation variances and intra-class variability.

3.2 Motivation

The present work is inspired from two different facts, the first one being the inability in using standard cryptographic techniques in biometric protection schemes (some ad-hoc investigations have been done but with limited success). Almost every cryptographic primitives (e.g. Hash functions, Block ciphers etc.) have been thoroughly analyzed and theoretically studied by the research community, thereby providing some strong constructions with well-defined bounds. On the other hand, the security of biometric protection scheme is mainly evaluated by the abstract notion of entropy or sometimes by minimum entropy associated with the stored data. However, for biometric based systems, this measure considerably differs between theory and practice. Cryptographic schemes are not directly applicable to biometric systems since their identification/verification processes are not deterministic but prone to fluctuations or variations. This proposed model tries to bridge this gap and provide some strong cryptographic guarantees in the biometric domain.

The second aspect of this work is related to the recognition accuracy rates of the biometric system. As mentioned previously, there exists a trade-off between the various requirements desirable in a typical biometric protection scheme. This trade-off is particularly apparent between the security levels promised by the system and its usability (i.e. recognition accuracy and speed). Especially, the area of cancelable biometrics provides the necessary properties of *irreversibility* and *unlinkability* quiet efficiently, but causes a significant degradation in the performance of the system

[14]. Other alternative technique like the *fuzzy vault* requires a multiple numbers of iterations to authenticate a genuine user, which increases the operational time of the biometric system. The objective of this work is to present a cancelable scheme which preserves the original recognition accuracy rates while simultaneously operating in reasonable session time.

3.3 Model Development

This section presents in details the construction of the framework by theoretically describing each module.

3.3.1 System Modules

As stated previously, the complete framework is divided into six distinct blocks or modules. These are stated as follows.

3.3.1.1 Minutiae Extraction

Minutia points in the framework can be extracted by any standard fingerprint feature extraction technique, but preferably in the ISO/IEC 19794-2 standard format [135]. Since this work is concerned with the security aspects of fingerprint templates, emphasis on any particular method for feature extraction is not required (an accurate feature extraction algorithm helps in maintaining high recognition accuracy rates

for the system, which is very desirable). Moreover, extraction of any helper data along-with the minutiae points is not required. Importantly, this step eliminates any potential overheads (e.g. leakage of information) that may be associated with any helper data. At the end of this module, the minutia points are represented by a triplet of parameters (x, y, θ) where each term has it's usual semantics. Thus for a total of N minutiae points, the feature set M is obtained as -

$$M = (x_i, y_i, \theta), \quad \text{where} \quad i = 1, 2, ...N$$

3.3.1.2 Pre-alignment

A pre-alignment operation is necessary for making the minutiae points invariant to any rotational and translational variations introduced during their extraction from the fingerprint images (illustrated in Figure 3.2).

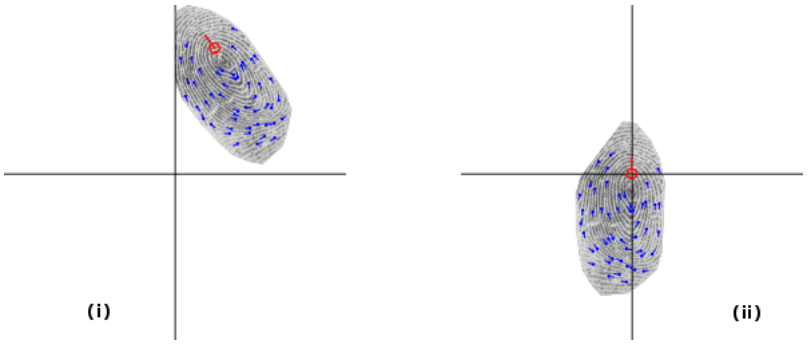


FIGURE 3.2: The problem of fingerprint alignment:(i)before alignment (ii)after alignment

The pre-alignment technique which has been implemented in this framework was first proposed in [136]. The main principle behind this scheme's working was estimating

a directed reference point from a fingerprint, and then representing the minutiae w.r.t. the resulting Cartesian co-ordinate system using linear algebra techniques. The directed reference point (i.e. a reference point with a definite direction) of a fingerprint was estimated by approximating a coordinate on a region where the orientation field locally looks like the orientation field near the core of a tented arch. Such a field of a tented arch was modeled by the quadratic differential model presented in [137]. The ultimate objective of the study was to find the rotation and translation of such a tented arch model so that it approximated an orientation field of the fingerprint. The core of the fitted model was used as the reference point's location information and the direction of the longitudinal axis was assumed to be its associated direction. For experimental purposes, the orientation fields were estimated using gradient method following the description in [127]. The resulting estimation technique was controlled by a tuple of six parameters -

$$(d_{core}, d_{delta}, \rho, \sigma, \lambda, R)$$

where, d_{core} = distance of a core which is placed on the tented arch model's longitudinal axis to the origin, d_{delta} = distance of a delta which is placed on the tented arch model's longitudinal axis to the origin, ρ = distance of the fitted arch from the core where orientation measurements with the highest weight are taken into account, σ = standard deviation of the Gaussian function which was used for fitting the tented arch with the orientation field of the fingerprint, λ = a real parameter

associated with the orientation field of an arch when modeled by the complex function $\psi(z) = \lambda^2(z^2 - R^2)^2$, and R = another real parameter of the complex function $\psi(z)$.

The minutiae templates (M) were subsequently shifted (pre-aligned) w.r.t. to the directed reference point estimated from these parameters. The pre-aligned minutiae templates M' are represented as -

$$M' = (x'_i, y'_i, \theta'_i), \quad \text{where} \quad i = 1, 2, ...N$$

3.3.1.3 Quantization

Quantization of minutiae points accounts for the variability between two fingerprint samples from the same user. In this work, a rigid regular hexagonal grid was first constructed over the fingerprint minutiae points and subsequently each minutiae point was associated to the grid point which best approximated it. Finally, the minutia points were encoded as functions of this grid point and a parameter which controlled the number of quanta used for quantization of the minutia's angle. This process is diagrammatically illustrated in Figure 3.3.

Formally speaking, let the hexagonal grid be represented by the set of grid points $\phi = \{\phi_1, \phi_2, ...\phi_r\}$, where r is the total number of grid points. The equidistance spacing between these grid points is specified by δ. This grid covers an entire region of size $H \times W$ where all the pre-aligned minutiae points lie. For fingerprint images,

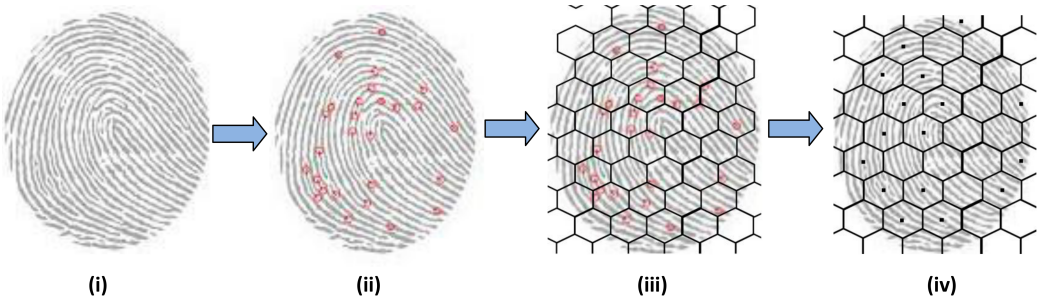


FIGURE 3.3: Quantization of minutiae points:(i)fingerprint image (ii)extracted minutiae (iii)overlapped hexagonal grid (iv)quantized minutiae

the values of H and W represent the dimensions of the image. Let $\phi_i, (1 < i < r)$ represent the grid point which best approximates the location of a minutiae point $m', (m' \in M')$. Also, let s represent a parameter controlling the number of quanta used for quantizing the minutia angles. Let,

$$i' = \left\lfloor \frac{\theta'}{2\pi s} \right\rfloor$$

The minutiae m' is encoded in the hexagonal grid as-

$$quant(m') = i + r.i'$$

The optimum value of the parameter δ (which specifies the pattern of the hexagon grid) was empirically determined, as stated in subsequent sections.

3.3.1.4 Salting

The notion of *revocability* is enforced in the framework by the addition of cryptographic salts. In general, salting is a processing step before the application of any hashing function. The process of salting involves concatenating a certain amount of randomly generated data to the original information, thereby enabling the same data to be hashed into a completely different output every time. On detection of any anomaly in the database, the old templates can be revoked and new templates can be generated by altering the salt value. Additionally, salting the templates helps in preserving the privacy of the users, thereby eliminating the risks of cross-linking or correlation based attacks.

A proper application of salts must follow three essential guidelines regarding their size, generation procedure and re-usability. Specifically, the length of the salt should be ideally equal to the length of the output digest of the succeeding hash function (as a rule of thumb). For this framework, a salt of length 256 bits is concatenated since the SHA-256 hashing function has been used in the next module. Secondly, the hash must be generated not in any deterministic way, but in a purely random method. In this work, the salts have been incorporated through ISSAC, a fast and secure cryptographic random number generator [1]. Let the features obtained after concatenating d bits salt to the original data be represented by the set T. Thus,

[1] http://www.burtleburtle.net/bob/rand/isaacafa.html

$$T = \{T_1, T_2, ...T_n\} \quad \text{where} \quad T_i = quant(M_i')\|d, 1 \leq i \leq n$$

Here $\|$ is the concatenation operator.

One of the most significant usefulness of salting is the reduction in the FAR of the recognition system. More specifically speaking, if there is no collision and the salts are non-repeated, the matching scores for any impostor equals zero. The reason for this is attributed to the properties that each enrolled subject is associated with a specific salt value, and as such the hash digests for all the subjects are unique. For this framework, the salt must not be stored in the database alongside the protected templates. Instead, they should be kept on separate token based systems like smart cards (which are provided to their respective users).

3.3.1.5 Hashing

Before discussing the importance of this module in the framework, a brief overview of general and cryptographic hash functions is presented. Hash functions are specialized mathematical functions $hash(m)$ having an arbitrary message as input. Importantly, they must satisfy the following two properties-

i The function is deterministic and 'easy' to compute (in terms of efficiency).

ii The function takes an input of arbitrary length while produces a fixed length of output d.

Formally speaking, a hash function is a function:

$hash : \mathcal{M} \rightarrow \mathcal{R}$, such that $\mathcal{M} = \{0,1\}^*$ is the message space and $\mathcal{R} = \{0,1\}^d$ is the range for some $d \geq 1$.

A cryptographic hash function is a subset of normal hash functions which follows some more addition properties such as -

iii **Pre-image resistance** - A hash function is pre-image resistant if, given a hash value h, it is hard to find the message m such that $h = hash(m)$.

iv **Collision Resistance** - A hash function is collision resistant if, given two messages m_1 and m_2, it is hard to find a hash h such that $h = hash(m_1) = hash(m_2)$.

v **Second Pre-image Resistance** - A hash function is second pre-image resistant if given a message m_1, it is hard to find a different message m_2 such that $hash(m_1) = hash(m_2)$.

The SHA-256 hash function has been utilized in this work. It is a part of the SHA-2 family of cryptographic hash functions developed by the NSA [138]. This hash function works on 32-bit words and produces a message digest of length 256 bits. Functionally, it is based on the Merkle-Damgȧrd construction [139] which repeatedly constructs collision-resistant cryptographic hash functions from collision-resistant one-way compression functions. The *irreversibility* property of a biometric

protection scheme is satisfied in this case by the *pre-image resistance* paradigm of the hash function. Specifically speaking, let the hash function be denoted by $\mathcal{H}$ and the final 256 bits outputs by the set O. Thus,

$$O = \{O_1, O_2, ...O_n\} \quad \text{where} \quad O_i = \mathcal{H}(T), 1 \leq i \leq n$$

Although the SHA-256 hash function has been specifically implemented in this model, other secure hash functions like RIPEMD-160 [140] or WHIRLPOOL [141] can also be used.

3.3.1.6 Matcher

This is the final module in the framework wherein a matching score is generated based upon the similarity between two sets of hash digests; one obtained during enrollment (O) and the other extracted during an authentication attempt (O'). The score ($Match$) is calculated as -

$$\begin{aligned} Match = J(O, O') = \frac{|O \cap O'|}{|O \cup O'|} = \frac{|O \cap O'|}{|O| + |O'| - |O \cap O'|} \\ = \frac{|O \cap O'|}{2n - |O \cap O'|} = \frac{|HD(O_i, O'_j) = 0|}{2n - |HD(O_i, O'_j) = 0|} \end{aligned} \tag{3.1}$$

Here, $i, j = \{1, 2, ...n\}$, J is the Jaccard similarity coefficient, HD is the hamming distance between two hashed strings and $|.|$ operator stands for cardinality of the

matching criterion. A matching decision is enforced by specifying a decision threshold (DT) on the matching score.

The complete proposed framework is diagrammatically illustrated in Figure 3.4.

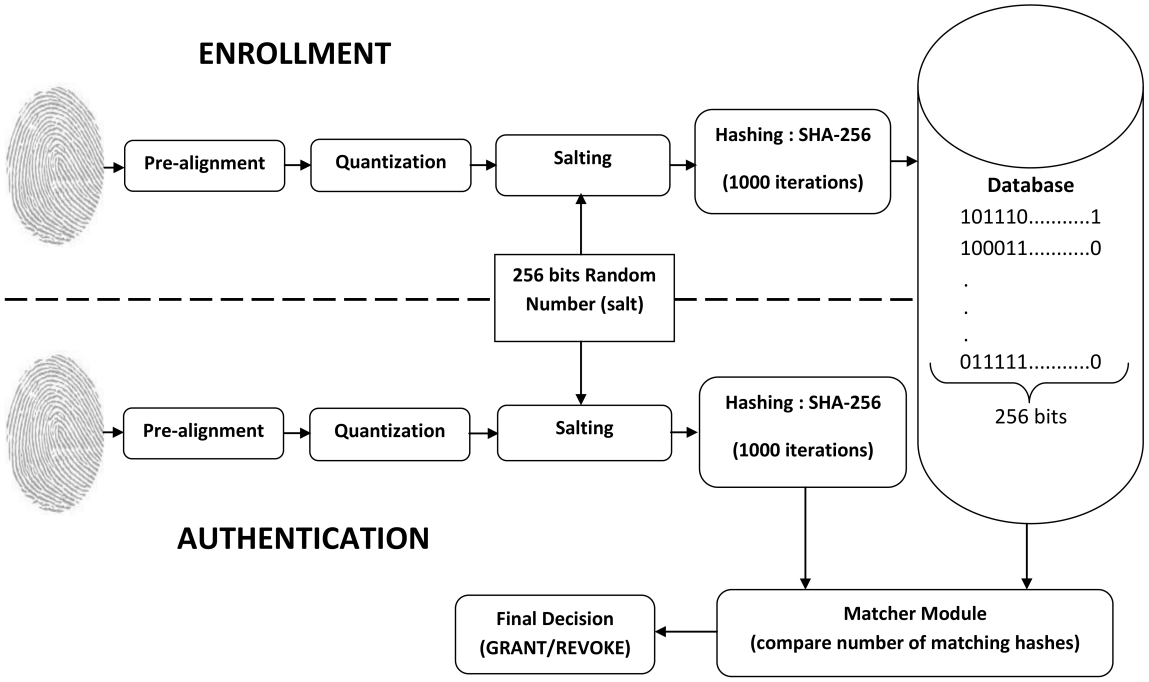


FIGURE 3.4: Proposed biometric template security framework.

3.3.2 Database Size

The size of the final database is very important as it provides a realistic estimate about the space requirements. According to the parameters of the proposed framework, required space for each enrolled individual equals to the size of each hash digest $\times$ no. of hash digests $= (d \times n)$ bits.

Thus total size the database equals to -

$$size = \frac{(d \times n)}{8} bytes$$

For empirical values of $d = 256$ (due to use of SHA-256) and $n = 30$ (variable), the value of $size \approx 960$ bytes. This size can be further reduced by the use of a hash function which produce digests of smaller size (e.g. RIPEMD-160 which produces a total size of ≈ 620 bytes).

3.3.3 Matching Complexity

Analyzing the matching complexity of the scheme provides a temporal estimate of the overall construction. This analysis is important since the usability of a scheme partially depends on the time taken during the matching process. The matching procedure is basically a multivalued function which compares every element of the querying feature hash digests with the stored hash digests. Thus for n number of such comparisons, the overall complexity of matching can be estimated as -

$$time = O(n^2)$$

Hence the matching complexity quadratically increases with the input size.

3.4 Theoretical Security Analysis

The security properties for any biometric template protection scheme can be understood by the scheme's resilience against any unauthorized attempt for learning any information about the stored templates. For analyzing the security of the proposed framework, some generalized attack scenarios are considered. From an adversary's point of view, his/her ultimate objective is to obtain some additional information about the enrolled users apart from the data which is already present in the public domain. In this case, it is assumed that the entire database is available to the adversary. This enables the adversary to get direct access to the hashed minutiae digests and other database specific parameters required during the pre-alignment and quantization steps. On the basis of these objectives, the following three attack scenarios along with potential attacks strategies can be devised.

3.4.1 Template Inversion

The adversary aims to expose the original fingerprint templates of the users in the template inversion based attacks. For obtaining the original form of the data, the adversary would require to invert the hash digests into their original input form. This operation directly corresponds to the *pre-image resistance* property of the implemented hash function. Ideally, a hash function should generate for any set of inputs, a set of outputs that is uniformly distributed over its output space. Thus,

if one counts the frequency of the number of 1 bits in its outputs, a binomial distribution $B(n, p)$ with p=0.5 should be observed. Let the number of bits produced by the hash function be denoted by d. Thus the brute force success probability for finding a pre-image from a hash digest becomes (denoted by $P[S_1]$) -

$$P[S_1] = \frac{1}{2^d}$$

For inverting all the hash digests for any particular user (assuming all the digests are independent), the final success probability becomes -

$$P[S_1] = \frac{1}{2^{nd}}$$

where n is the total number of encoded minutiae points of a user.

Since the SHA-256 produces a digest size of 256 bits, the brute force probability of finding the associated pre-image of any digest equals $\frac{1}{2^{256}}$. Till date, the best pre-image resistance attack was mounted in [142]. It employed bicliques by combining powerful techniques from differential cryptanalysis of block ciphers and hash functions. The attack itself reduced the complexity from $\frac{1}{2^{256}}$ to $\frac{1}{2^{255.5}}$ in 45 rounds (out of 64). Thus any practical pre-image attack on this hashing technique is almost impossible.

3.4.2 Cross-linking/Diversity

In the cross linking attack, an adversary tries to link the hashed values across multiple databases enrolling the same user, thus attempting to gain any additional information about the corresponding user. For normal transactional databases, this form of linking attacks is very harmful since it directly breaches the privacy of the users. The primary basis for performing this type of attack involves investigating the similarity between two transformed templates obtained from the same user. This type of situation in biometric systems is very common since a user usually enrolls in multiple biometric authentication systems. For the proposed model, the problem of finding same hash digests for multiple instances of a single user is associated with the *collision resistance* property of the hash function. For any hashing scheme producing d bits in the digest, an adversary requires on average $2^{d/2}$ evaluations using a birthday attack. The associated success probability becomes $\frac{1}{2^{d/2}}$. However this probability also comprises of the collisions associated with different users. Let's assume that there are x databases $(Db_1, Db_2, ...Db_x)$ in which a common user U has enrolled. Also, let the number of participants in the databases be $(\eta_1, \eta_2, ...\eta_x)$ respectively. So the probability that a hash digest (O) belongs to user U while linking all the databases becomes -

$$P[O \in U] = \frac{x}{\sum_{i=1}^{x} \eta_i}$$

Subsequently, the overall success probability becomes -

$$P[O' = O|O', O \in U] = \frac{1}{2^{d/2}} \times \frac{x}{\sum_{i=1}^{x} \eta_i}$$

Since linking all the minutiae points requires the linking of all the hash digests for a user, the final success probability evaluates to (denoted by $P[S_2]$) -

$$P[S_2] = \frac{1}{2^{nd/2}} \times \frac{x}{\sum_{i=1}^{x} \eta_i}$$

For SHA-256, collisions have been detected in 46 rounds (out of 64) with a complexity of 2^{46} [143]. As for now, the SHA-256 hash function is considered collision resistant against practical attacks.

3.4.3 Stolen Token Attacks

Stolen token attacks are based on the assumption that the adversary succeeds in obtaining the key providing cancelability. In this model, these keys are the database specific salt values which are concatenated prior to hashing. The strategy of the adversary for conducting such an attack would be to select all the possible minutiae feature values from the input space, concatenate the salt value with them and finally apply the hashing functions to generate sample output hash digests. Subsequently, the adversary can match the sample outputs with the real hash outputs in the database to filter out possible minutiae data. The success probability of this attack directly depends upon the input space since the adversary can efficiently try out all

the possible values from the input space (i.e. brute force) if the space is too small. In this case, the input space corresponds to the quantized minutiae values obtained during the hexagonal grid based quantization process.

Let the hexagonal grid covering a fingerprint image of dimension $W \times H$ be denoted by the set of grid points be $\phi = \{\phi_1, \phi_2, ...\phi_r\}$. Let r denote the total number of grid points and δ be the equidistance spacing between these grid points. Also, let n denote the number of minutiae points present in the fingerprint image, n_ϕ denote the average number of minutiae points present in each hexagonal grid [2] and k denote the number of hexagonal grids in which the n minutiae points are present ($k < r$; a majority of the hexagonal grids would remain empty). Thus the relation $n_\phi = \frac{n}{k}$ is obtained. The parameters n_ϕ and k depend on the spatial distribution of the minutiae points, which in turn is specific to the database used. Also, let the area of each hexagonal grid be represented by a_ϕ. Using simple geometrical constructions, it can be showed that -

$$a_\phi = \frac{\sqrt{3}\delta^2}{2}$$

Thus the total number of hexagonal grids covering the fingerprint image equals to -

$$r = \frac{W \times H}{a_\phi}$$

[2] We assume that equal number of minutiae points are present in each hexagonal grid

Now, the brute force strategy of an adversary for finding original minutia points within a grid would be to test all the points present in the grid. Thus the corresponding success probability equals to $\frac{n_\phi}{a_\phi}$. Consequently, the success probability of the adversary for accurately reconstructing all the n minutiae points from the k hexagonal grids equals to -

$$P[S_3] = \left(\frac{n_\phi}{a_\phi}\right)^k = \left(\frac{2 \times n_\phi}{\sqrt{3}\delta^2}\right)^k$$

As observable in the experimental section, the framework has been evaluated on the FVC 2002-DB1,FVC 2002-DB2, FVC 2004-DB1 and FVC 2004-DB2 databases. The values of the aforementioned parameters for the two databasess are shown in Table 3.2.

Parameter	**Symbol**	**FVC 2002-DB1**	**FVC 2002-DB2**	**FVC 2004-DB1**	**FVC 2004-DB2**
Width of image	W	388	296	640	328
Height of image	H	374	560	480	364
Equidistant spacing of grids	δ	25	29	25	25
Grid area	a_ϕ	541.26	728.38	541.26	541.26
No. of grids	r	268	228	568	221
No. of occupied grids	k	35	32	40	35
Success probability	$P[S_3]$	2.14×10^{-96}	2.54×10^{-92}	4.62×10^{-110}	2.14×10^{-96}

TABLE 3.2: Parameters of FVC databases for stolen key based attack scenario (considering $n_\phi = 1$)

Since the adversary's success probabilities are extremely small, the proposed framework provides considerable levels of security in case the cancelable keys (i.e. salts) get stolen.

3.5 Experiments and Results

This section provides details about the databases, experimental protocols and obtained results. Furthermore, the results are analyzed and compared with that of other techniques implementing cancelable schemes. The simulations of the framework were performed via the C++ security research libraries THIMBLE [3] and Crypto++.

3.5.1 Data Acquisition and Performance Metrics

3.5.1.1 Database

For evaluating the performance of the proposed system, experiments were conducted on four databases namely FVC 2002-DB1, FVC 2002-DB2 [144], FVC 2004-DB1 and FVC 2004-DB2 [145]. These databases were selected because they are in the public-domain and contain good quality of fingerprint images. Moreover, much of the prior biometric research works have been assessed on these databases. Each dataset contains 100 users and each user has 8 samples; hence there are 800 (100×8) fingerprint images in total. The fingerprint images for all the four databases were scanned using optical scanners at dpi of 500, 569, 500 and 500 respectively. However, the quality of fingerprints in the FVC2002 databases is comparatively much better than those in the FVC2004 databases. Sample images from these databases are presented in Figure 3.5.

[3] http://www.stochastik.math.uni-goettingen.de/biometrics/fileadmin/thimble

(a) 2002-DB1 (b) 2002-DB2 (c) 2004-DB1 (d) 2004-DB2

FIGURE 3.5: Sample fingerprint images from the FVC databases

3.5.1.2 Feature Extraction

According to previous discussions, the minutiae points extracted from the fingerprint images were in the standard 19794-2:2005 format [135]. Digital Persona's FingerJetFX OSE C++ library [4] was used for the extraction process. The FingerJetFX OSE application performs feature extraction on fingerprint image files in PGM (portable graymap) format and saves the fingerprint minutiae data in the ISO/IEC 19794-2:2005 format.

3.5.1.3 Matching Protocol

The standard FVC protocols were followed for testing the proposed framework. In all the databases, genuine scores computation was done by matching each sample of a user with the remaining seven samples of the same user, thereby generating a total of 2800 matching cases. Alternatively, the impostor scores were calculated by matching the first sample of each finger in the database against the first sample of

[4] http://digitalpersona.com/fingerjetfx

the remaining fingers in the database; thereby generating a total of 4950 impostor matching cases. Care was taken such that symmetric comparisons were excluded while matching for both the genuine and impostor cases.

3.5.1.4 Performance Measures

The results obtained in this paper are evaluated in terms of FAR, FRR and EER. Apart from these metrics, the overall performance is visualized by plotting the corresponding ROC curve. Besides the ROC curve, another performance illustration namely genuine-impostor distribution is also used to justify the recognition accuracy. A clean separability between the genuine and impostor distributions indicates better performance while a strong overlapping region between them implies poor performance.

3.5.2 Framework Analysis

In this section all the experimental results are systematically presented with their detailed analysis. The authentication phase was simulated while considering both the *plain verification* and *stolen token* scenarios.

3.5.2.1 Quantization via Hexagonal Grids

Firstly, the effects of varying the number of hexagonal grids (quantization levels) on the overall system performance are investigated. The inter-grid distance parameter δ

was chosen as the varying parameter since the number of hexagonal grids imposed on the fingerprint images is directly determined by δ. Conceptually speaking, a larger value of δ would result in bigger equidistance spacing between successive hexagonal grids, which would ultimately result in lower number of hexagonal grids. Alternatively, smaller values of δ correspond to lower inter-grid spacing and consequently greater number of hexagonal grids.

Different values of δ were selected and the effects on the performance in terms of EER were observed. The values for δ were altered from 20 to 35 with an interval of 5 (i.e. $\delta = \{20, 25, 30, 35\}$). In agreement with other similar works, the experiments were conducted under the stolen-token scenario. Table 3.3 displays the performance for all the four databases while incorporating the specific δ values.

Database	**Subset**	**No. of grids/quantization levels** (r)	**Inter-grid distance** (δ)	**EER(%)**
FVC 2002	DB1	419	20	7.5
		269	**25**	**5.8**
		187	30	6.5
		137	35	7.2
	DB2	479	20	6.6
		307	25	5.42
		213	**30**	**5.3**
		157	35	6.1
FVC 2004	DB1	444	20	19.5
		284	25	18.1
		198	30	17.12
		145	**35**	**15.8**
	DB2	345	20	16.1
		221	25	15
		154	**30**	**14.5**
		113	35	14.7

TABLE 3.3: Tuning of quantization levels for obtaining optimum performance (same key scenario).

As observable, the best recognition performances are obtained for $\delta = 25, 30, 35, 30$

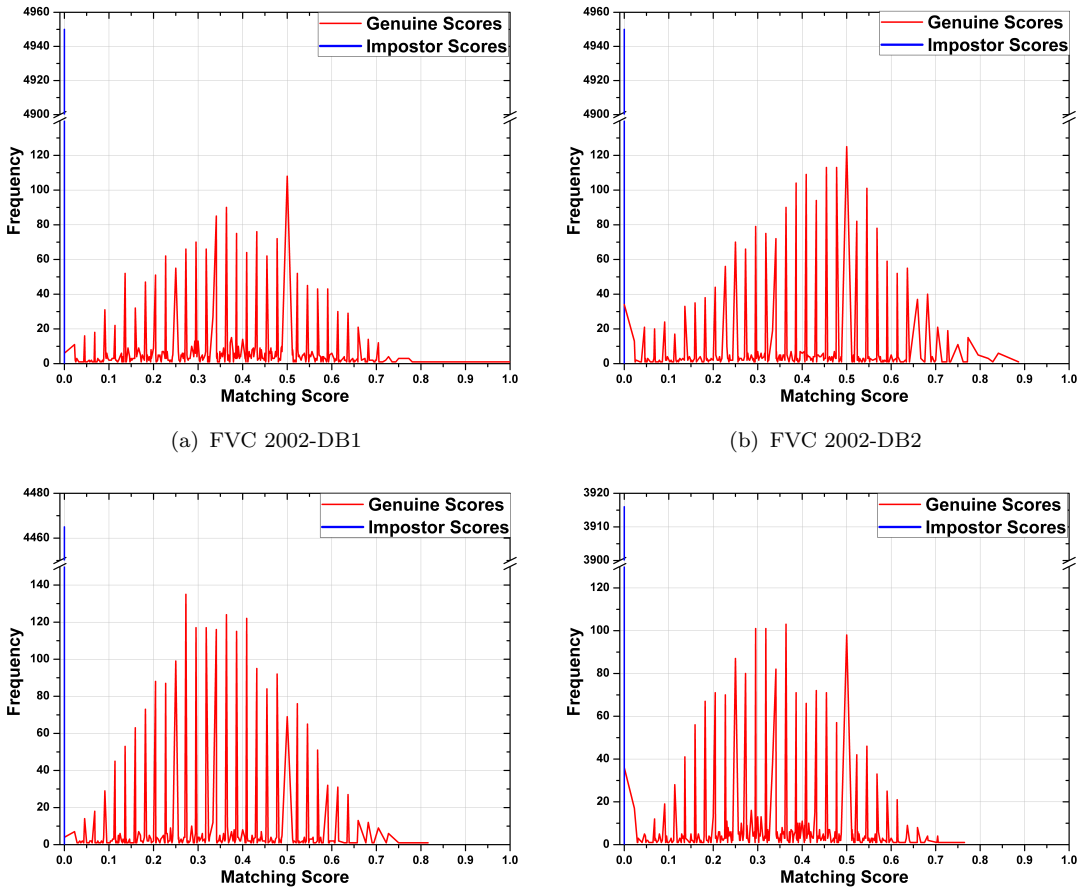
corresponding to the 2002 FVC-DB1, DB2, 2004 FVC-DB1, DB2 databases respectively. Subsequently in this work, these specific settings have been used for further experiments. Importantly, it is observed that there is no fixed value of δ which optimizes the recognition performance for all the databases. Hence it can be safely concluded that the value of δ depends on the intrinsic properties of a database, like scanning resolution and image dimensions.

3.5.2.2 Plain Verification Scenario

The plain verification scenario corresponds to the situation when each enrolled user presents their respective keys (salt values) during the verification phase. This is the default working case for any biometric recognition system. The genuine-impostor distributions for all the four databases are shown in Figure 3.6. The experiment shows that the results are almost ideal, which is denoted by a clear separability between the two distributions. It is noticeable that the impostor scores of this scheme always remained zero under this scenario, which consequently resulted in extremely low values of EERs (less than 1%). Table 3.4 shows the EERs corresponding to the four databases.

Database Name	FVC 2002-DB1	FVC 2002-DB2	FVC 2004-DB1	FVC 2004-DB2
EER (%)	0.2	0.8	0.1	0.9

TABLE 3.4: EERs for FVC databases under the plain verification scenario.



(a) FVC 2002-DB1

(b) FVC 2002-DB2

(c) FVC 2004-DB1

(d) FVC 2004-DB2

FIGURE 3.6: Genuine-impostor distributions under the plain verification scenario.

3.5.2.3 Stolen Token Scenario

The stolen token scenario represents the special case when it is assumed that an impostor possesses the tokens (salts) of the genuine users. Using the stolen credentials, the impostor later tries to circumvent the biometric system by enforcing a false positive error. The same salt value is utilized for all users to simulate the stolen-token scenario. In this context, a specific salt is generated in advance and subsequently assigned to all the users. The genuine-impostor distributions for the four databases under such a constraint is shown in Figure 3.7. Expectantly, the

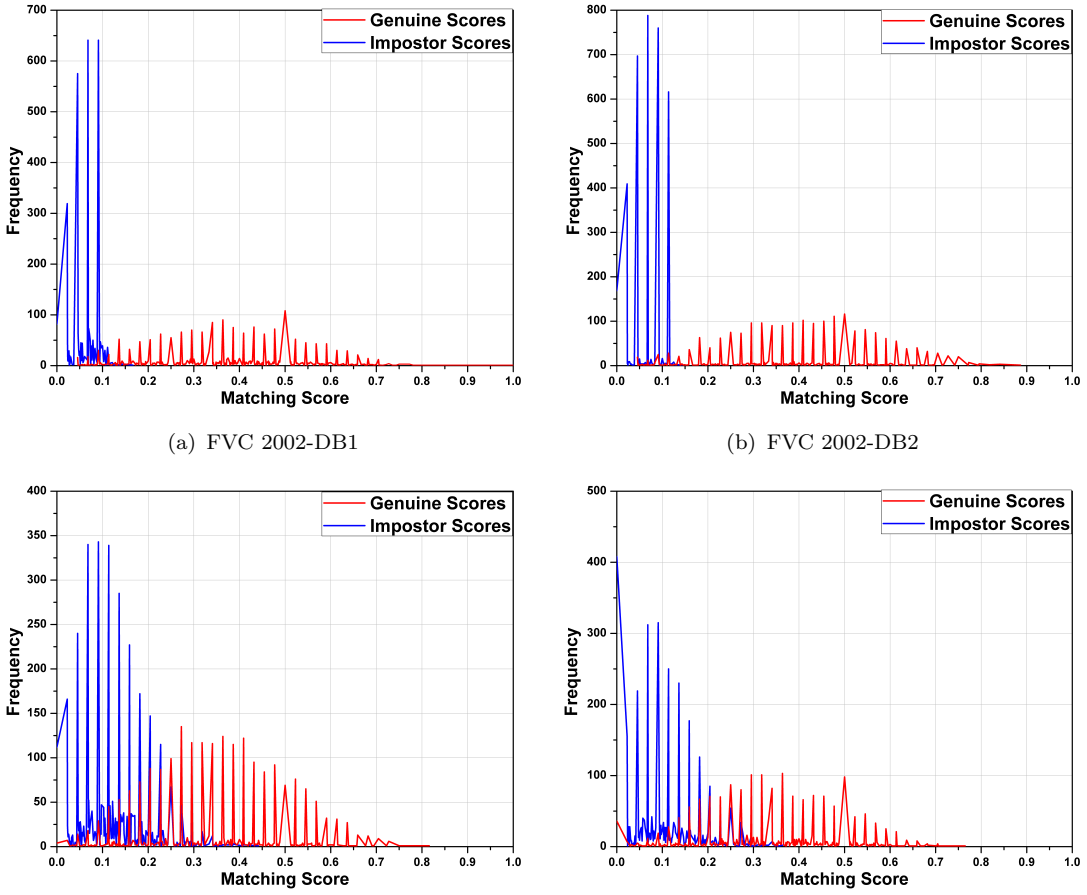
overlapping regions between the two scores are considerably greater than that in the plain verification scenario. This performance degradation is normal since an adversary gains the additional knowledge of genuine user specific keys in such a case.

The computed EERs are depicted in Figure 3.8 (via dotted lines) corresponding to the four databases. Accordingly, EERs of 5.8%, 5.3%, 15.8% and 14.5% were obtained for the 2002 DB1, 2002 DB2, 2004 DB1 and 2004 DB2 databases respectively. Noticeably, these EER values have already been optimized by utilizing appropriate quantization levels. The performance degradation in FVC2004 databases can be attributed to the low fingerprint image quality in FVC2004 as compared to FVC2002. The overall performance of the scheme under the stolen token scenario is also illustrated in Figure 3.9 and Figure 3.10 by plotting the ROC curves [5]. Under FVC2002, the scheme performs better for the DB2 dataset in comparison to its DB1 counterpart, whereas under FVC2004, it performs almost similarly for DB1 and DB2.

3.5.2.4 Performance Comparison

In this section a performance metric of the proposed scheme is compared with other contemporary biometric cancelable schemes for fingerprints. Although based on separate design strategies, all these schemes achieve the same objective of generating cancelable features from fingerprint data. The comparison itself is performed by

[5]Since it has been explicitly stated that the error rates among different FVC competitions should not be compared with each other, the ROC curves for FVC2002 and FVC2004 are depicted separately



(a) FVC 2002-DB1

(b) FVC 2002-DB2

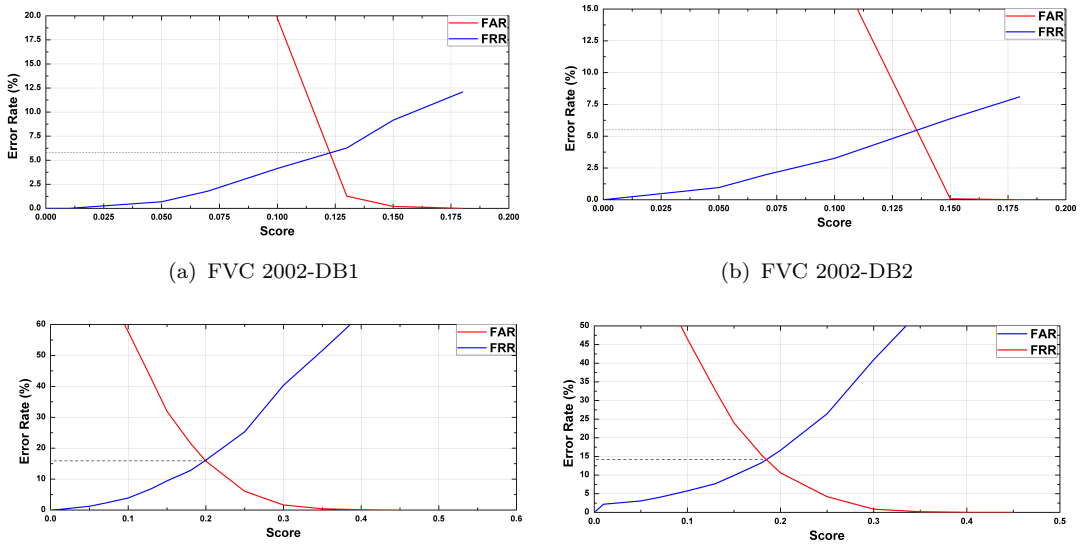
(c) FVC 2004-DB1

(d) FVC 2004-DB2

FIGURE 3.7: Genuine-impostor distributions under the stolen token scenario.

observing the EER reported for these methods. Table 3.5 depicts the EER for various schemes along with their principal design features. In agreement with previous works, this scheme is compared with the other methods under the stolen token scenario.

As observable from the tabular comparison, the proposed scheme attains EERs comparable to other state-of-the-art methods. Although this scheme underperforms as compared to [113] and [114], it offers better performance when compared to [116], [110] regarding the FVC-2002 databases and [115] regarding the FVC-2004



(a) FVC 2002-DB1 (b) FVC 2002-DB2

(c) FVC 2004-DB1 (d) FVC 2004-DB2

FIGURE 3.8: EER for the FVC databases under the stolen token scenario

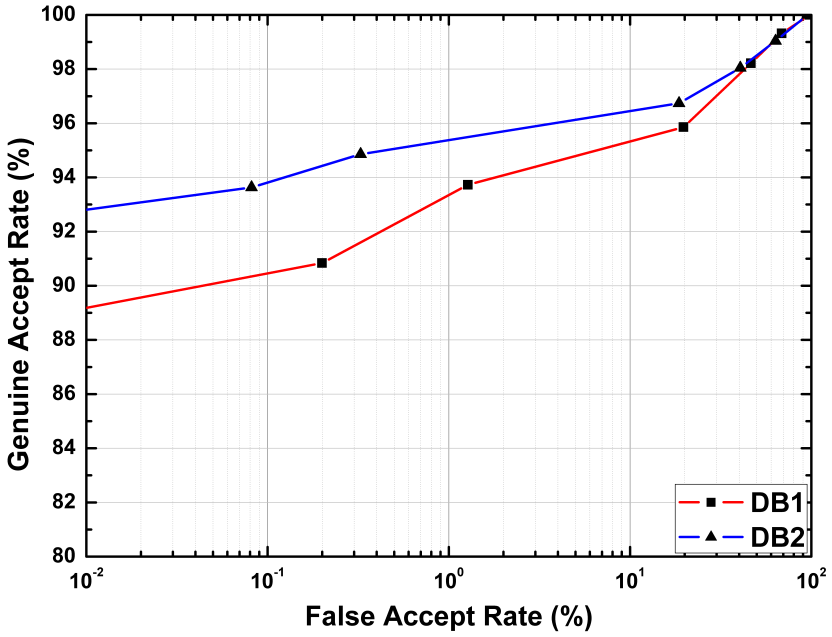


FIGURE 3.9: ROC curves for FVC 2002 under the stolen token scenario

databases. Other methods like [104],[111] provide similar performance measures to that of this technique, specially for the FVC-2002 DB2 database.

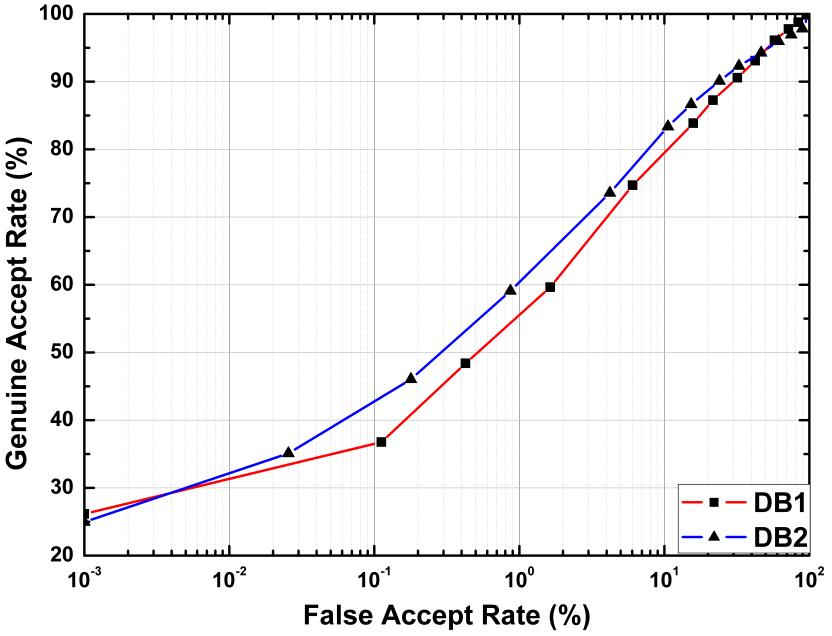


FIGURE 3.10: ROC curves for FVC 2004 under the stolen token scenario

Ref.	Key Features	FVC 2002		FVC 2004	
		DB1-A	DB2-A	DB1-A	DB2-A
[116]	Pair-polar co-ordinate based template design	9	6	15.76	11.64
[104]	Inter minutiae distance vector based graphs	4.26,2.27	5.06,3.79	-	-
[111]	Densely infinite-to-one mapping (DITOM)	3.5	5	-	-
[110]	Polar grid based 3-tuple quantization	5.19	6.94	-	-
[113]	Local Voronoi neighbor structures	3.38	0.59	16.52	14.88
[115]	Randomized graph based Hamming embedding	4.36	1.77	24.71	21.82
[114]	3D array based quantization and Delaunay triangulation	3.96	2.98	12.17	13.29
[102]	Dynamic random projection (RP)	-	4.53	-	-
Proposed	**Cryptographic hash functions**	**5.8**	**5.3**	**15.8**	**14.5**

TABLE 3.5: Performance comparison via EER(%) for various fingerprint based template protection schemes under the stolen token scenario.

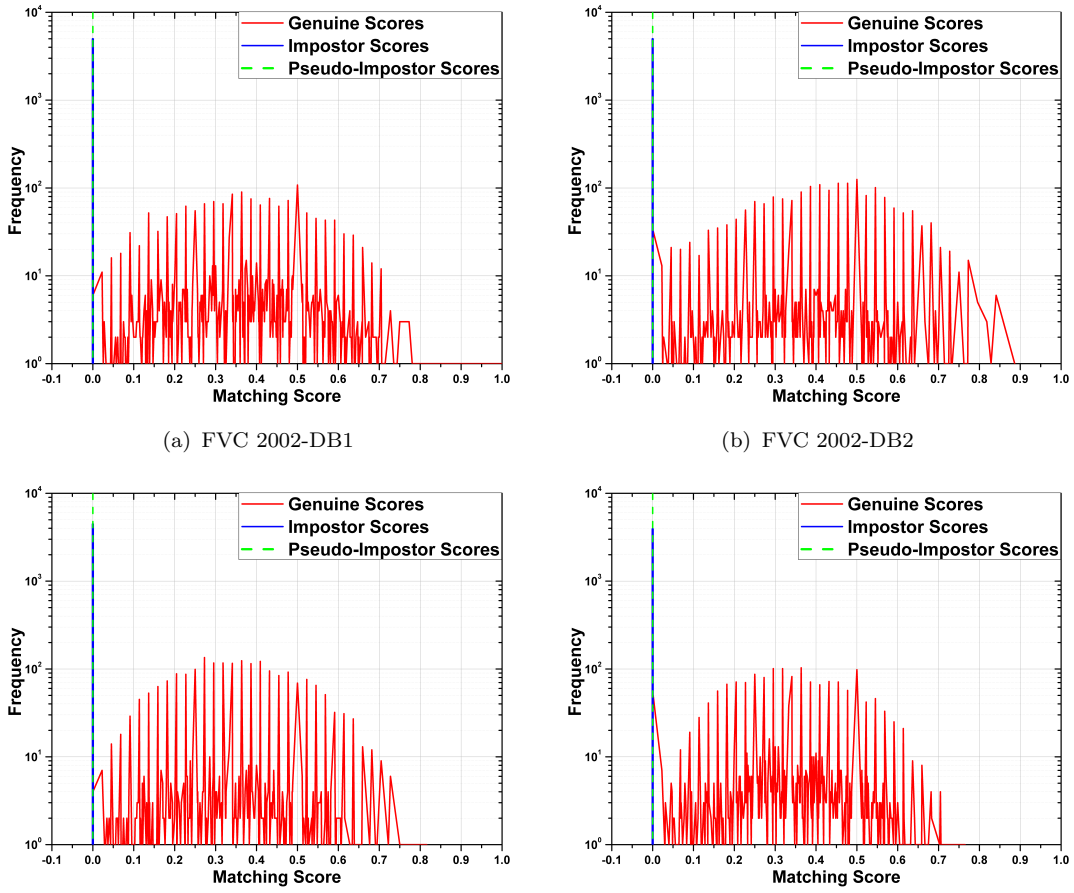
3.5.3 Evaluation on Security Criteria

In this section, the security notions of *revocability* and *diversity* of the scheme are empirically vindicated.

3.5.3.1 Revocability

The property of revocability can be verified in biometric systems by virtue of generating *pseudo-impostor* distributions. In this particular scenario, a number of different templates are generated from a single biometric trait and are subsequently matched with the enrolled template of the trait. The claim of revocability can be justified if (1) impostor distribution and pseudo-impostor distribution are overlapped; (2) genuine distribution and pseudo-impostor distribution has clear separability [110].

For this purpose 100 hash digests from a single fingerprint image were generated by concatenating 100 distinct salts and subsequently compared with its hash digest obtained during enrollment. This process was repeated for 100 different fingerprints (the 1st fingerprint of the 100 users were taken), thereby generating 10,000(100×100) pseudo-impostor comparisons in total. The pseudo-impostor distribution along with the genuine and impostor distributions for the four databases are shown in Figure 3.11. As noticeable from them, the pseudo-impostor distribution completely overlaps with the impostor distribution corresponding to all the databases. All the pseudo-impostor comparisons result in zero matching scores, thereby enabling the perfect overlapping with the impostor distribution. As a consequence, the EERs obtained for this case is identical to that obtained in the plain verification scenario (presented in Table 3.4). Significantly, this positive result can be accredited to the *collision resistance* property of the used hash functions and the randomness of the salt generation procedure. Thus it can be concluded that even if multiple hash digests are



(a) FVC 2002-DB1

(b) FVC 2002-DB2

(c) FVC 2004-DB1

(d) FVC 2004-DB2

FIGURE 3.11: Genuine,impostor and pseudo-impostor distributions under the plain verification scenario

generated from the same fingerprint, they do not match with the enrolled digest. The revocability criterion can accordingly be achieved.

3.5.3.2 Diversity

Diversity states that the template generated from a biometric trait must not allow crossing-matching among other templates generated from the same trait, thus ensuring the user's privacy. Hence a user can enroll for multiple applications by

utilizing his same biometric trait every time. For simulating this situation, 100 different hash digests corresponding to a fingerprint were generated (by utilizing 100 different salts) and subsequently a 1-to-all cross matching was performed among them. Hence for each fingerprint, $\frac{100\times99}{2} = 4950$ test cases were generated. This entire process was repeated for 100 distinct fingerprints, thereby producing a total of 4,95,000(4950×100) matching cases. The average matching scores for these cases regarding the four databases are shown in Table 3.6.

Database Name	FVC 2002-DB1	FVC 2002-DB2	FVC 2004-DB1	FVC 2004-DB2
Matching Score	0	0	0	0

TABLE 3.6: Average matching scores for FVC databases under the *diversity* protocol.

The results clearly show that all the hash digests generated from the same biometric data were different from each other, thus resulting in zero matching scores. This assertive result was consistent for all the four databases, thus vindicating the diversity property of the scheme. Similar to the previous case, this result is a direct consequence of the cryptographic hash function's *collision resistance* property, as well as the randomness incorporated during generation of salts.

3.6 Conclusion

In this chapter, various problems associated with fingerprint based biometric security schemes have been addressed and accordingly a secure solution has been proposed. The proposed scheme attempts to fulfill all the desirable properties of

such a scheme including *unlinkability*, *diversity*, *irreversibility* and *usability*. Unlike previously suggested cancelable schemes, this method does not result in the degradation of performance of the overall system for two reasons - firstly due to robust pre-alignment and quantization of fingerprint minutiae points prior to transformation and secondly performing an injective mapping between these points and the transformed templates (bounded by the collision resistance property of the hash function). Moreover in contrast to fuzzy vault schemes, the proposed scheme performs the matching of queried users in a single attempt by generating a similarity score.

The pivotal design of the model revolves around the implementation of a cryptographically secure hash function; specifically SHA-256 in this case. The inherent properties associated with hash functions like *pre-image resistance* and *collision resistance* caters to the majority of security requirements in the scheme. Additionally, pre-processing modules like directed reference point estimation based alignment and hexagonal grid based quantization mitigates standard biometric issues like spatial variances and intra-subject variability. The ensemble of all these features results in a model which has strong theoretical security bounds, as well as satisfactory recognition accuracy rates.