**Abstract**—This paper presents feature vector based fuzzy vault method for cryptographic key generation from fingerprint data. The proposed method not only outputs high entropy keys, but also conceals the original biometric data such that it is impossible to recover the biometric data even when the stored information in the system is open to an attacker. Simulated results have demonstrated that constructed Fuzzy Vault Crypto Biometric Key Based on Fingerprint Vector Features addresses a replacement of minutiae fingerprint vault representation, where the vector features are used to form the lock set. The size of the fuzzy vault and the degree of the underlying polynomial are needed for reaching the reliable investigated bounded key. This results in a high unlocking complexity for attackers with an acceptable unlocking accuracy for the legal users.

Key words: - Biometric, Fingerprint, Features, Fuzzy vault, Cryptography, Security.

I. INTRODUCTION

Biometric authentication system offers greater security and convenience than traditional personal verification systems. Biometrics is about measuring unique personal features [1]. It has the potential to identify individuals with a high degree of assurance, thus providing a foundation for trust. Biometric signals are also harder to copy or steal, and cannot be forgotten or lost. On the other hand, cryptography concerns itself with the ensuring the authenticity of information. Unfortunately, cryptographic security is conditioned by an authentication step typically based on long pseudo random keys, which are almost impossible to remember. As a result, cryptosystems commonly rely on user generated passwords, which are easy to memorize, in order to release the pseudo random keys. Biometric cryptosystems combine cryptographic security with biometric authentication [2]. The merging of biometric and cryptography has been introduced as an effective means to address the convenient and security issues of privacy enhancing technologies with respect of personal data protection. It is also introduced to solve cryptographic keys management problems, because all cryptographic algorithms fully depend on the assumption that the keys will be kept in absolute secrecy. It intends to bind a cryptographic key with the user’s biometric information in a manner to meet the distortion tolerance and discrimination requirements. Therefore, Crypto-Biometric merging can be done broadly at two different modes [3]: (1) loosely-coupled mode (biometric key release), the biometric matching is decoupled from the cryptographic part. Biometric matching operates on biometric templates, if they match; cryptographic key release from it is secure location, e.g. a server or smart card. (2) tightly-coupled mode (biometric key generation), biometric and cryptography are merged together at a much deeper level, i.e. the biometric signals are monolithically bounded to the keys. The matching at this level can effectively take place within cryptographic domain. Cryptographic construct, i.e. Fuzzy Vault, is an example of biometric crypto bounded mode.

II. PREVIOUS WORK

To overcome the integration of biometric into cryptosystem, a number of approaches were introduced in the literatures. In [4], Juels and Wattenberg present the first binding crypto-biometric system called fuzzy commitment, in which cryptographic key is de-committed using biometric data. The committed value will be extracted when the fuzziness value close to the original enrolled value. The fuzzy commitment scheme is based on the use of error correcting codes, considers binary strings where the similarity is measured by Hamming distance. In fuzzy commitment, the de-commitment key is built from an enrolled fingerprint, as a set of minutiae positions. These yield two shortcomings. First, it does not allow key modifications, such as re-ordering, and addition/deletion of an element in the key, although such modifications are frequent in real life. Second, the security proof of this scheme holds only if enrolled fingerprint is uniformly distributed, which is not the case in reality. In order to overcome these drawbacks, Juels and Sudan propose a fuzzy vault scheme (FVS) [5]. This scheme designed to work with biometric features, (e.g., minutiae in fingerprints). These features are represented as \((x, y, \theta)\) of ridge ending or bifurcation, where \((x, y)\) is minutiae coordination and \((\theta)\) is the angle of the associated ridge. Figure (1) depicts the operation of FVS. The idea behind FVS is to encode a secret \(s\) as a polynomial \(p\) of degree \(d\) using the Reed-Solomon encoding scheme (polynomial constructor block) and evaluate the polynomial on all elements of biometric template points. The user then chooses a large number of random false points which do not lie on the polynomial \(p\) (point generator block).
Figure 1 Fuzzy vault system block diagram

The total points (real polynomial points and chaff, i.e. false points) are used to form the vault $V$. The chaff points conceal the real points lying on $p$ from an attacker. The secret key can be retrieved from the vault $V$ by providing another query template. Using polynomial reconstructor block to format a query unordered set. If the query set overlaps substantially with reference set, then user can identify many points in $V$ that lie on $p$. If sufficient number of points on $p$ can be identified, an error correction scheme can be applied to exactly reconstruct $p$ and thereby decode the secret key. If reference points do not overlap substantially with query points, it is infeasible to reconstruct $p$ and the authentication is unsuccessful. The scheme fuzziness come from the variability of biometric data and that dose not affect on the secret key retrieval from vault.

Based on fuzzy vault scheme, Clancy et al. [6] proposed a fingerprint vault using multiple minutiae location sets per finger (based on 5 impressions of a finger), they first find the canonical positions of minutia, and use these as the elements of the set $(A)$. They added the maximum number of chaff points to find $(R)$ that locks $(k)$. However, their system inherently assumes that fingerprints (the one that locks the vault and the one that tries to unlock it) are pre-aligned. This is not a realistic assumption for fingerprint-based authentication schemes. Jain, Uludag, Pankanti, Ross, and Prabhakar further developed the fuzzy vault scheme for fingerprint minutiae [7-10]. The authors achieved FRR = 21% and FAR = 0 (i.e., less than one in 10000) under ideal conditions: fingerprints were manually pre-aligned and rotated, and the correspondence of the minutiae across images was manually established. Clancy et al. [6] simulated the error-correction step without actually implementing it. They found that 69-bit security (for False Accept Rate (FAR)) could be achieved with a False Reject Rate (FRR) of 20-30%. Note that the cited security translates to $2^{-69} = 1.7 \times 10^{-21}$ FAR. Further, FRR value suggests that a genuine user may need to present her finger multiple times to unlock the vault. Uludag and Jain [10] used lines based fingerprint minutiae representation to design fuzzy vault system, but it was without the actual implementation. Their work suffers from complexity and alignment problems. It differs from Clancy system in way that both location and angle of minutia are used to extract lines for forming the templates. Nagar and Chaudhury proposed another modified fuzzy vault scheme [11] which is used within asymmetric crypto system. A zero error rate (i.e., FRR=FAR=0) is reported for a small database of fingerprints. However, these unrealistic numbers were likely obtained because all attackers had different secret random masks assigned and, naturally, their templates did not get a match with a legitimate template. All previous reviewed approaches assumed that the source of reference data, i.e. fingerprints (the one that locks the vault and the one that tries to unlock it) are pre-aligned, which is not a realistic assumption in practical fingerprint authentication systems. Chung et al. [12] proposed a geometric hashing technique to perform alignment in a minutiae-based fingerprint fuzzy vault but still has the problem of limited security. That is, the maximum number of hiding points (chaff points) for hiding the real fingerprint minutiae is limited by the size of the fingerprint sensor meanwhile the size of the fingerprint images captured and the possible degradation of the verification accuracy caused by the added chaff minutiae. All approaches in [6, 10, 12] assumed the number of chaff points was 200. Lee et al [13] proposed both the automatic alignment of fingerprint data and higher security by using a 3D geometric hash table. A number of chaff points for proposed approach were more than in previous approaches in two times, as well as a complexity of cracking the proposed system was very high. A processed extracted feature vector is used in this paper as a replaceable base for fuzzy vault scheme.

III. PROPOSED APPROACH

Fuzzy vault based on fingerprint features vector, or so called FingerCode [14, 15] is constructing a bounded bio crypto key. The idea behind using FingerCode as a replaceable seed of minutia set fuzzy construct is the distinguishability of extracted FingerCode where it is reasonably stable. A FingerCode is composed of an ordered enumeration of the feature extracted from the local information contained in each image sector in cropped images. A feature vector is the collection of all features in each filtered image. These features capture both the global pattern of ridges and valleys and the local characteristics. The FingerCode scheme of feature extraction distinguishes the region of interest in cropped image, i.e. cropped images into [16X16], [32X32] and [64X64] surrounding the image reference point. The scheme is divided into two stages: preprocessing and feature extraction stages Figure (2).

A. Preprocessing

The preprocessing stage contains three main steps: Center Point Determination, Cropping, Sectorization and normalization the region around the reference point.

1) Center Point Determination

Center point location is done to find the point of most curvature by determining the normal's of each fingerprint - ridge, and then following them inwards towards the centre. We use the following steps to determine fingerprint reference point

- Image Noise reduction using 2-D Gaussian lowpass filter.
of 60 sectors (12 wedges × 5 bands). Another reason for sectorization is for normalization purposes. Each sector is individually normalized to a constant mean and variance to eliminate variations in darkness in the fingerprint pattern, due to scanning noise and pressure variations. And all the pixels outside of the sector map is considered to be one giant sector. This will yield in an image that is more uniform. The following is the equation which we used for normalization of each pixel. We used a constant mean $M_0$ and variance $V_0$ of 100. $i$ is the sector number, $M_i$ is the mean of the sector, and $V_i$ is the variance of the sector.

$$N_i(x, y) = \begin{cases} 
M_0 + \frac{V_0 [I(x, y) - M_i^2]}{V_i} & \text{if } I(x, y) \geq M_i \\
M_0 - \frac{V_0 [I(x, y) - M_i^2]}{V_i} & \text{otherwise} 
\end{cases}$$

(2)

2) **Image Cropping**

The image is then cropped into three options of cropping images centered around this pseudo-centre point.

3) **Sectorization and Normalization**

The cropped fingerprint image is divided into 5 concentric bands centred around the pseudo-centre point. Each of these bands has a radius of 20 pixels, and a centre hole radius of 12 pixels. Thus, the total radius of the sectorization is 223 pixels. Each band is evenly divided into 12 sectors. The centre band is ignored. This process of sectorization is done because of the feature extraction section. 6 equi-angular Gabor filters will be used which will align with the 12 wedges formed by the bands. In other words, each sector will capture information corresponding to each Gabor filter. The centre band is ignored because it has too small an area to be of any use. The radius of the sectorization was chosen to avoid the effects of circular convolution in applying a Gabor filter. Thus we have a total of 60 sectors (12 wedges × 5 bands). Another reason for sectorization is for normalization purposes. Each sector is individually normalized to a constant mean and variance to eliminate variations in darkness in the fingerprint pattern, due to scanning noise and pressure variations. And all the pixels outside of the sector map is considered to be one giant sector. This will yield in an image that is more uniform. The following is the equation which we used for normalization of each pixel. We used a constant mean $M_0$ and variance $V_0$ of 100. $i$ is the sector number, $M_i$ is the mean of the sector, and $V_i$ is the variance of the sector.

$$N_i(x, y) = \begin{cases} 
M_0 + \frac{V_0 [I(x, y) - M_i^2]}{V_i} & \text{if } I(x, y) \geq M_i \\
M_0 - \frac{V_0 [I(x, y) - M_i^2]}{V_i} & \text{otherwise} 
\end{cases}$$

(2)

B. **Feature Extraction**

1) **Gabor Filtering**

We then pass the normalized image through a bank of Gabor filters. Each filter is performed by producing a 33x33 filter image for 6 angles (0, π/6, π/3, π/2, 2π/3 and 5π/6), and convolving it with the fingerprint image. Spatial domain convolving is rather slow, so multiplication in the frequency domain is done; however, this involves more memory to store real and imaginary coefficients. The purpose of applying Gabor filters is to remove noise while preserving ridge structures and providing information contained in a particular direction in the image. The sectorization will then detect the presence of ridges in that direction. The Gabor filter also has an odd height and width to maintain its peak center point. The following is the definition of the Gabor filter [15]:

$$G(x, y, \theta) = e^{-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2}\right)} \cdot \cos(2\pi f x')$$

(3)

where $x' = (x \cos \theta + y \sin \theta), y' = (-x \sin \theta + y \cos \theta)$ are rotated coordinates.

2) **Feature Vector**

After we get the 6 filtered images, we calculate the variance of the pixel values in each sector. This will tell us the concentration of fingerprint ridges going in each direction in that part of the fingerprint. A higher variance in a sector means that the ridges in that image were going in the same direction as is the Gabor filter. A low variance indicates that the ridges were not, so the filtering smoothed them out. The resulting 360 variance values (6 × 60) as the feature vector of the fingerprint scan. The following is the equation for variance calculation. $P_{i\theta}$ are the pixel values in the $i$th sector after a Gabor filter with angle $\theta$ has been applied. $P_{i\theta}$ is the
mean of the pixel values. \( K_i \) is the number of pixels in the \( i \)th sector.

\[
V_{i\theta} = \frac{\sum_{K_i}(F_{i\theta}(x, y) - P_{i\theta})^2}{K_i}
\]  

(4)

The result will be three vector features for cropped iamges. A concatenation value of these vectors will formulate final used feature (5).

\[
V = V_1 \parallel V_2 \parallel V_3
\]  

(5)

Where this vector will be used as true date point and will replace minutiae points' feeder in Fuzzy vault scheme [5, 6, 9, 10] construction to generate the needed vault, Figure (3), algorithm (1), and different chaff point distibution Figure (4) algorithm (2).

Algorithm (1) chaff point generation
Inputs: Feature Vector points (true points) Threshold distance
Output: TP (total points) – Array containing all true
and chaff points.

Algorithm:

\[
TP \leftarrow \text{(number of chaff points, thresh, true points)}
\]

Point randomly chosen from the coordinates \((0..2^{16})\)

Do while conditions

If it within distance from previously selected points {

Then discard it

If not {

Select it

\[
TP = TM \cup CP
\]

Return

End

Algorithm (2) Vault construction
Inputs: Injected secret key (SK); Total points (TP)
Output: \( V_{(TP)} \) (Vault total points);

Algorithm

\[
V_{(TP)} \leftarrow \text{SK encoded message as coefficients of polynomial, Galois field array from TP}
\]

Do compute

test for vault authentication

Galois field array of SK in the coordinate’s \( (0..2^{16}) \)

Galois field array of TP in the coordinate’s \( (0..2^{16}) \)

\[
V_{(TP)} = \text{Evaluate polynomial value of encoded SK & TP}
\]

Return \( V_{(TP)} \)

End

IV. RESULTS AND CONCLUSION

Proposed approach implemented on the FVC2004-DB1 [17], a public domain database with 800 images (100 fingers 8 impressions each finger), (cropped into 256x256 sizes, 500 dpi resolutions). True points taken on base of concatenating extracted vectors from three cropped images, where the average of extracted minutiae is 21 points while chaff points were chosen to be 300 points. Point threshold distance adapted to 6. According to the Berlekamp Welch error correcting code theory with the polynomial degree, \( d \). The condition in equation is \( 2k + d < m \) which means to successfully decode the finger vault, the number of impostor points must satisfy \( k < (m-d)/2 \), where \( m \), is the total number of the original input points. The constructed vault result will be tested under all reconstruction parameters, like polynomial degree, minimum distance of point distribution, and vault complexity. Figure (5) shows the relationship between polynomial degree and vault attack complexity where the used extracted feature points are 21 points while chaff points were chosen to be vary from 100 to 300 points. While Figure (6) shows the relationship between chaff points, minimum distance and release ability of locked key. We set the minimum distance to satisfy the following rules: chaff points cannot be placed to close to real points, no reason to place chaff points next to each others at any distance less than minimum distance, because the attacker can immediately ignore them as unlikely candidates.

REFERENCES


Figure 3. Feature vector points distribution

Figure 4. Total points (original and chaff) distribution

Figure 5. The attack complexity varies according to the degree of polynomial for different size of chaff point

Figure 6. The relationship between the key releasability and the minimum distance of chaff point for different chaff points size.