International Journal of Electronics and Communication Engineering (IJECE)
ISSN(P): 2278-9901; ISSN(E): 2278-991X
Vol. 3, Issue 2, Mar 2014, 1-8

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

SECURE BIOMETRIC CRYPTOSYSTEM USING FINGER CODE

SHANTHINI K. SREENIVASAN

Department of Applied Electronics and Instrumentation, Government Engineering College, Kozhikode, Kerala, India

ABSTRACT

Biometric cryptosystems combine cryptographic security with Biometrics-base authentication. These authentication systems are more user-friendly and more reliable since biometric data cannot be lost, forgotten, or guessed. When cryptography is combined with biometrics, the digital signature created by a user is linked with him/her more accurately. A secure fingerprint authentication method is proposed here. In this method, a non invertible transformation is used to generate a finger code from a set of minutiae points. This is linked with a key to produce the helper data. The key is reproduced only when the user submits his biometric for authentication. Since the helper data does not reveal any information about the biometric or the key, it is quite difficult for an attacker to procure any of these. In order to accommodate the variability in biometric data, error correction is applied. The proposed algorithm has been tested with DB1 (FVC 2000) dataset which shows the feasibility of the technique in practical usage.

KEYWORDS: Biometric Cryptosystem, Finger Code, Feature Vector, Minutiae Points

1. INTRODUCTION

The user authentication with cryptographic system requires a key. A highly secure cryptographic system demands that the size of the ret key to be larger which is almost impossible to remember. Hence a user is asked to select a password for encrypting the cryptographic key. But there are two major issues related to this. First because of the problem of remembering various passwords the user may choose a simple password which is easy to crack. Secondly, cryptographic system may not be able to discriminate between an attacker and a legitimate user since there is no direct connection between the user and password.

Today, the most important problem in the use of biometrics is the template security. In the context of biometric authentication, one way to deal with this problem is by designing a robust and one-way transformation. There are two requirements for this transformation. It should be able to map all possible noisy versions of the original biometric data to the same, unique output value since the error tolerance is essential when biometric data are considered. Secondly, the transformation should be non-invertible. That is, given the output, it should be infeasible to find an input that result in that output. Such a transformation should not reveal much information about the original biometric data. Cryptographic hashes are one-way functions, but they cannot be used due to variability in the biometric data.

Biometric cryptosystem offers solution to these problems by introducing a new mechanism for key security in which a cryptographic key is secured by using a biometric. Biometric based authentication consists of two stages: enrolment and verification. During *Enrolment* the features are extracted from the biometric collected from the person. Then a template is derived which is encoded and stored in a database.

In the *verification* stage, the stored template is matched with the live biometric sample presented by the user. Authentication services can be more reliable and convenient by the incorporation of biometrics into cryptosystem. One of

the approaches is known as biometric key binding where the key is linked with the biometric during enrolment which will be released later during verification upon a successful biometric authentication. This scheme has advantage that the key and biometric data does not depend on each other and hence even if the key is ever compromised, it can be easily modified at a later stage.

Biometric authentication systems are to be designed in such a way that False Acceptance Ratio (FAR) is very low close to zero. This is necessary to ensure authenticated access. Hence design methods of Biometric authentication systems should concentrate on minimization of False Rejection Ratio (FRR) while keeping FAR almost zero. A biometric authentication method based on fingerprints is proposed in this paper. Detailed analysis shows that the proposed method has good performance in comparison with the other methods available in literature.

2. RELATED WORK

Clancy et al. [1] used the fuzzy vault for building and analyzing a fingerprint based authentication scheme. The technique of fuzzy vault had been first introduced by Juels and Sudan [2]. In Clancy's work, minutiae coordinates are extracted from a fingerprint image. A locking set is formed from these minutiae locations. Through a polynomial reconstruction a ret key is derived from this. Non related chaff points are also added to increase the unlocking complexity. The ret key can be recovered by a Reed-Solomon code only if there is a high degree of overlap between the input and testing fingerprint. But unfortunately the system shows 30% false rejection rate. A similar method proposed by Uludag et al. [3] for fingerprints was also an implementation of fuzzy vault. However the security was reduced to roughly 35 bits due to simplifications they made to [1] and empirical testing showed a FRR of 21% at 0% FAR.

Hao and Chan proposed an authentication scheme based on handwritten signatures in [4]. Signature features were extracted from dynamic information such as velocity, pressure, altitude and azimuth. Each feature was quantized in to bits using feature coding and a unique binary string was formed from this. They could obtain 40 key bits. FRR is about 28% and FAR is 1.2%.

Hao et al. [5] implemented a practical iris cryptosystem. Their scheme used 2048-bit string called an IrisCode [6]. A concatenated Error Correction Code is employed which is a combination of Reed-Solomon code and Hadamard code based on a study of the error pattern in the iris data. For a key size of 140 bits, they reported FRR =0.47% FAR= 0%. However, the experiments were conducted with iris dataset which is free from degrading factors such as illumination, focusing variations, occlusion of eyelashes, etc. The paper also shows that if an attacker knows correlation pattern of the Iris code, there is a dramatic reduction in security to 44 bits, which makes it difficult for practical usage.

Yagiz Sutcu et.al. [7], proposed a geometric transformation for securing the minutiae based fingerprint templates. This scheme employs a robust one-way transformation that maps geometrical configuration of the minutiae points into a fixed-length code vector. This representation enables efficient alignment and reliable matching.

3. SCHEME PROPOSED BY YAGIZ SUTCU ET. AL

In this scheme, the set of minutiae points extracted from a fingerprint are represented as a set coordinates denoted as

$$F_i=\{(x_1,y_1)_i, (x_2,y_2)_i,\ldots\ldots\ldots\ldots (x_n,y_n)_i\} \quad (1)$$

Where n is the number of extracted minutiae points and index i denotes the user. These data points are then

mapped onto a circle, which encompasses all the data points, via a one-to-many transformation. Hence, the minutiae points extracted from a fingerprint are represented by a fixed length code vector, called fingerprint code. With the proposed scheme, the matching of two fingerprints depends on the distance between the corresponding fingerprint codes. Furthermore, due to nature of the deployed transformation, determining minutiae positions of the fingerprint from a given fingerprint code is extremely difficult. Basic steps of the proposed transformation are as follows:

Extraction: Extract the set of minutiae points from the fingerprint image.

Mapping: Calculate the centroid point of the minutiae set F_i and draw a circle centered at this centroid point with radius *R*. The equation of the circle is:

$$(x-x_{c,i})^2+(y-y_{c,i})^2=R^2 \tag{2}$$

Where $(x_{c,i}, y_{c,i})$ is the centroid point of the user i which is calculated as

$$y_{c,i}=\left(\sum_{k=1}^{n} y_k\right)/n \tag{3}$$

$$x_{c,i}=\left(\sum_{k=1}^{n} x_k\right)/n \tag{4}$$

For every pair of minutiae point in the set F_i, if the distance between that pair of points is greater than a predefined threshold value, T, draw a line which passes through those two minutiae points and determine the two points of intersection between the circle and the line.

Quantization: Organize intersection points on the circle into bins according to their position on the circle where each bin is generated by partitioning the circle into arcs of Δ degrees.

Code Generation: The number of points in each bin will be concatenated together to create the fingerprint code of size 360/Δ

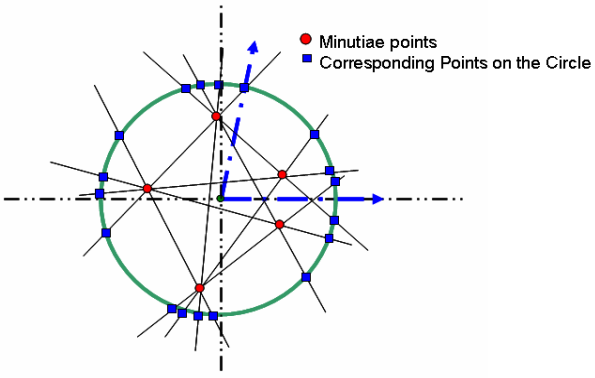

Figure 1: Minutiae Mapping Proposed by Yagiz et al

Testing of this algorithm is done on synthetic fingerprints. For each fingerprint image, noisy versions are created each of which consists of 20 randomly generated minutiae points. As a measure of closeness between the template and query fingerprint, mean absolute error between the template code and the query fingerprint code is used.

The overall performance of the scheme is worse in the case where minutiae addition/deletion is considered. Real fingerprints often contain noise which will mask some of the genuine minutiae or may introduce spurious minutiae. Insertion or deletion of even a single minutiae point will drastically change the number of lines drawn through the pair of minutiae points. Since each line has two intersection points on the circle the corresponding change in the finger code will be also large. Similarly, larger distance threshold values eliminate relatively more minutiae pairs (less number of lines to consider) which in fact means throwing away some useful information that the original fingerprint posses, thereby decreasing the performance of this scheme. Hence we modified this scheme so that it works for real fingerprint images.

4. PROPOSED SCHEME

4.1 Feature Vector Generation

- Extract the minutiae points which are a set of coordinates.
- Calculate the centroid using equations (3) and (4). Draw lines which pass through each minutiae point and the centroid, and determine the point of intersection between the circle and the line. Then do quantization and code generation as done in section 3.
- Obtain the finger code which can be represented as :

$$C_i = \{ f_1, f_2, \ldots\ldots\ldots f_p \} \text{ where } p=360/\Delta \quad (5)$$

Where f_i are the intersection points in each bin.

- Assume that there is a random p by q matrix R_i(called randomization matrix) associated with each user. Compute $C_i \times R_i$ .So final feature vector is given by

$$V_i = \{ v_{i1}, v_{i2}, \ldots\ldots\ldots\ldots\ldots v_{iq} \} \quad (6)$$

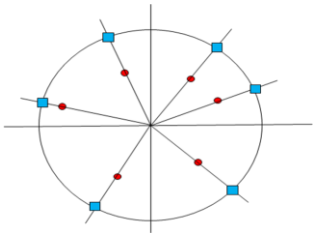

Figure 2: Minutiae Mapping for Proposed Scheme

4.2 Enrollment

During the enrollment first the fingerprint image is captured and minutiae points are extracted. The major steps in minutiae extraction [8] are image enhancement, normalization, binarization, thinning, and minutiae extraction. For image enhancement the method proposed by Hong et al [9] is used. These minutiae points are then mapped by a one-way transformation and a finger code is generated. Thus a feature vector V_0 is generated from the input fingerprint image produced by the user as described in section 4.1. The feature vector V_0 is replaced with its equivalent binary representation. A random key k_0 is generated as a string of random bits. The key k_0 is then encoded with Reed-Solomon (RS) code to get a pseudo key k_0'. The pseudo key k_0' will be locked by EXO Ring it with the user's feature vector V_0 to obtain bio code b_0. It is called a locked code, because by itself it cannot be used to deduce either the finger code or the biometric key.

$$b_0 = k_0^{'} \oplus V_0 \tag{7}$$

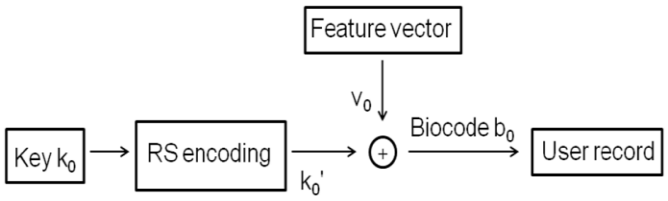

Figure 3: Key Binding in Enrollment Process

The bio code will be stored as part of the user record. The key k_0 is encrypted using bits from the bio code and is hashed to get an identification code id_0. This is also stored in the user record. Subsequently the key must be securely erased.

Reed-Solomon codes are *non-binary cyclic* codes with symbols made up of *m*-bit sequences, where *m* is any positive integer having a value greater than 2. The Reed-Solomon code is denoted as RS (*n, k, t*), where *k* represents the number of blocks before encoding and *n* represents the number of blocks after encoding. The *t* is the number of error blocks that can be corrected. Reed-Solomon codes achieve the *largest possible* code minimum distance for any linear code with the same encoder input and output block lengths. For Reed-Solomon codes, the code minimum distance is given by

$$d_{\min} = n - k + 1 \tag{8}$$

The coding must be able to correct the differences between error bits of finger codes for the same finger, but unable to correct the differences between different fingers.

4.3 Verification

Verification process will successfully retrieve the key for a legitimate user. The key bits are decoded in the key retrieval stage and they are verified in the key validation stage. During the verification phase, the user presents her fingerprint for feature vector generation. As in enrollment, feature vector generation consists of minutiae extraction followed by mapping of the minutiae points to create finger code which will be used for creating feature vector V_1. The feature vector V_1 is used to unlock the key. The unlocking process is executed by EXO Ring binary representation of V_1 with the bio code b_0 from the user record. This operation will produce the pseudo key k_1'.

$$k_1^{'} = b_0 \oplus V_1 \tag{9}$$

The pseudo key k_1' is then decoded with RS code to output the biometric key k_1.The newly derived key k_1 is encrypted and hashed using the same bits as that used for enrollment which in turn produces an identification code id_1. If id_1 matches the stored hash id_0, then the derived key is correct. Otherwise, the key will be assumed false and rejected.

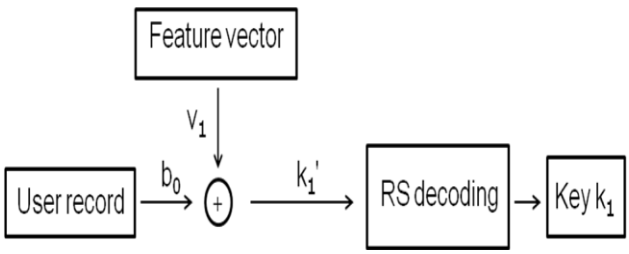

Figure 4: Key Retrieval in Verification Process

5. RESULTS

Fingerprints from FVC 2000 database (DB1) are used for the experiments. The proposed method has been implemented and the results were analyzed in MATLAB. The original image is of size 300x300. It is cropped to a size of 200x200. For the original image shown in figure 5, the enhanced and resized image is obtained as in figure 6. Figure 7 shows the extracted minutiae points. This includes the extreme and spurious minutiae points. Hence post processing is done in order to remove these unwanted minutiae points. Spurious minutiae points results because of the poor quality of the image which could not be improved by the enhancement. Sampling window technique [8] is used to extract the genuine minutiae points as in figure 8. Image cropping will result in ridge terminations along the boundaries which results a lot of extreme minutiae points. These extreme minutiae points are removed by defining a region of interest (ROI) as the portion of the image avoiding the 10 pixels on all the four boundaries. Once ROI is defined, it is multiplied with the image and minutiae points which fall in the dark regions of ROI are filtered out as shown in figure (9). Finally the genuine minutiae points are obtained as in figure 10.

Figure 5: Original Image

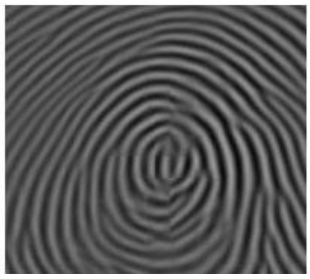

Figure 6: Enhanced and Cropped Image

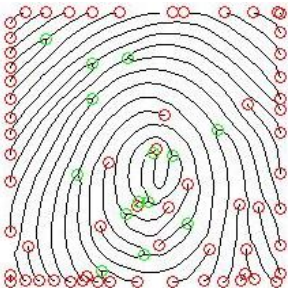

Figure 7: Extracted Minutiae

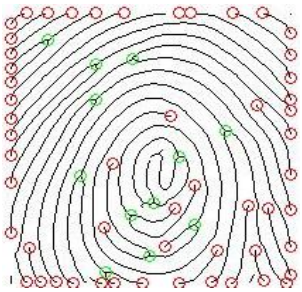

Figure 8: Spurious Minutiae Removed

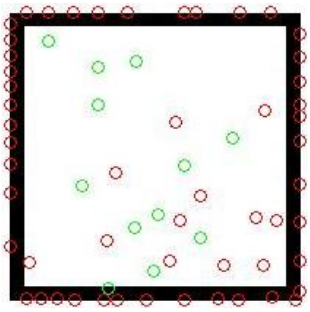

Figure 9: Region of Interest

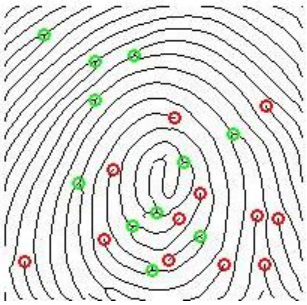

Figure 10: Genuine Minutiae

The finger code obtained for image is shown below. The circle is divided in to arcs of 45˚. i.e., total there are 8 bins. The bins are counted in the anti clockwise direction. The radius R is chosen as 150.

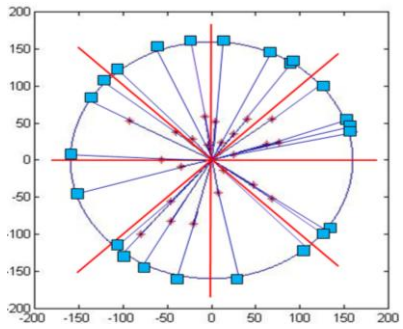

Figure 11: Minutiae Mapping for Image in Figure 5

Finger code = [4 4 3 3 1 4 2 2]

Implementation results of Biometric cryptosystem using Finger code are as given in table1. The key size is 28 bits.

Table 1: Implementation Results of Biometric Cryptosystem Using Finger Code

Genuine Acceptance Rate (GAR)	81.66%
False Rejection Rate (FRR)	18.33%
False Acceptance Rate (FAR)	0%

6. CONCLUSIONS

A secure biometric cryptosystem is proposed in this paper. The proposed scheme provides a simple and efficient solution to the template security problem. It is difficult to invert the proposed transformations to estimate the original locations of the minutiae points. The information loss due to binning conceals the exact locations of the intersection points on the circle. This method has added advantage that the user does not need to remember any passwords. A user independent random key is linked with biometric which is also stored through a noninvertible transformation. This table is maintained in the database for subsequent verification purposes. Error control coding is applied for reducing the variations in fingerprint data. A new sample of the fingerprint image will retrieve the key bits during verification. The false rejection rate (FRR) for this method is found to be 18.33% at false acceptance rate (FAR) 0%. These results are at par with the performance results of Biometric systems available in literature, which are explained in detail in section 2. The system has translation invariance of 10 pixels and rotation tolerance of 6 ˚.

REFERENCES

1. T.C. Clancy, N. Kiyavash and D.J. Lin, “secure smart card- based fingerprint authentication,” Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Application, WBMA 2003.
2. A. Juels and M. Sudan, “A fuzzy vault scheme,” Proceedings of IEEE International Symposium on Information Theory, 2002.
3. U. Uludag, S. Pankanti, and A. K. Jain, "Fuzzy Vault for Fingerprints," in Audio- and Video-Based Biometric Person Authentication, pp. 310-319, 2005.
4. F. Hao, C.W. Chan, “Private key generation from on-line handwritten signatures,” Information Management & Computer security, Issue 10, No. 2, pp. 159–164, 2002.

5. F. Hao, R. Anderson, J. Daugman. “Combining crypto with biometrics effectively”, IEEE Transactions on Computers, Vol. 55, No. 9, pp. 1081-1088, 2006.
6. J. G. Daugman, “High confidence visual recognition of personals by a test of statistical independence”, IEEE Trans. Pattern Anal. Mach. Intell., vol.15, no.11, pp.1148–1160, Nov. 1993.
7. Y. Sutcu, H. T. Sencar, and N. Memon. “A geometric transformation to protect minutiae-based fingerprint templates”. In SPIE International Defense and security Symposium, 2007.
8. Raymond Thai, “Fingerprint Image Enhancement and Minutiae Extraction”, Technical Report, The University of Western Australia, 2003.
9. L. Hong, Y. Wan, and A. Jain, "Fingerprint image enhancement: Algorithm and performance evaluation” IEEE Trans. Pattern Analysis and Machine In tell., vol. 20, no. 8, pp. 777--789, Aug. 1998.