Biometric Key Binding: Fuzzy Vault Based on Iris Images

Youn Joo Lee[1], Kwanghyuk Bae[1], Sung Joo Lee[1], Kang Ryoung Park[2], and Jaihie Kim[1]

[1] Department of Electrical and Electronic Engineering, Yonsei University, Biometrics Engineering Research Center
{younjoo,paero,sungjoo,jhkim}@yonsei.ac.kr
[2] Division of Digital Media Technology, Sangmyung University, Biometrics Engineering Research Center
parkgr@smu.ac.kr

Abstract. Recently, crypto-biometric systems have been studied for solving the key management problem of cryptographic systems and protecting templates in biometric systems at the same time. The fuzzy vault system is a well-known crypto-biometric system. We propose a new method of applying iris data to the fuzzy vault. Our research has following two advantages and contributions. First, in order to solve the variation problem of the extracted iris features, we introduce a pattern clustering method. Second, in order to produce unordered sets for fuzzy vault, we use the iris feature extraction algorithm based on ICA (Independent Component Analysis). Experimental results showed that 128-bit cryptographic keys as well as the iris templates were secure with the fuzzy vault scheme.

Keywords: Fuzzy Vault Scheme, Pattern Clustering, Independent Component Analysis (ICA).

1 Introduction

In general, cryptographic systems suffer from the key management problem [1], which refers to dealing with the storage of cryptographic keys and the secure generation of these keys. To overcome these problems, current cryptographic systems have stored keys in storage devices such as smartcards, computers or servers, to be released only by password-based authentication [1]. Therefore, the security level of cryptographic keys has depended on how robust a user's password is to brute force attacks [2] and how robust a user's stored keys are to physical attacks. Passwords are generally short and simple so that users can memorize them easily. Hence, password-based authentication always involves the threat of passwords being cracked by brute force attacks. Also, passwords may be shared, lost or forgotten. These problems can be solved by using biometric-based authentication. Some reasons for this are because biometric features are generally difficult to be copied, shared and distributed, and they usually require users to present at the time and point of authentication [1]. However, if stored keys can be released from storage devices by using simple biometric

S.-W. Lee and S.Z. Li (Eds.): ICB 2007, LNCS 4642, pp. 800–808, 2007.

matching, there still remains the risk of the keys and biometric templates being compromised by physical attacks. For example, if an attacker strikes a storage device, stored keys and templates can be readily compromised. To overcome these problems, a crypto-biometric system is needed, which merges cryptographic keys with user's biometric data, using cryptography. In crypto-biometric systems, keys and templates are combined and then stored in storage devices. Therefore, attackers cannot obtain keys without knowing the specific user's biometric data, so the keys and biometric templates can remain secure. The fuzzy vault scheme, as proposed by Juels and Sudan [3], is well known as a form of cryptography that binds cryptographic keys and biometric templates.

In this paper, we focus on applying iris data to a fuzzy vault scheme. We propose a method of extracting iris features suitable to fuzzy vault and then implementing our system. To extract invariant iris features, we used an iris feature extraction algorithm based on Independent Component Analysis (ICA) [4-5] and a pattern clustering technique. Our fuzzy vault system was achieved by combining iris data with a 128-bit Advanced Encryption Standard (AES) key [2].

The remainder of this paper is organized as follows: section 2 describes works related to the fuzzy vault scheme. In section 3, we discuss how to extract iris features that can be applied to the fuzzy vault scheme as well as the procedures necessary to implement the fuzzy vault based on the iris data. Experimental results are presented in section 4, and conclusions are drawn in section 5.

2 Related Works

Juels and Sudan [3] first proposed a fuzzy vault scheme which encrypted and decrypted secret information securely, using a fuzzy unordered set. When encrypting the secret data, these researchers first generated a polynomial (p) by encoding secret data (for example, by using the secret data (which corresponds to secret key) as the coefficients of the polynomial). Next, they projected the components (which corresponds to biometric data) of the unordered set as x-axis coordinates on this polynomial (p) and produced genuine point pairs $(x_i, p(x_i))$, $i = 1,2,\ldots,n$, where n refers to the size of the unordered set (X) on the polynomial (p). Then, chaff points which did not exist on the polynomial (p) were generated in order to protect the genuine point set $\{(x_i, p(x_i)), i = 1,2,\ldots,n\}$. Finally, they created a vault which was a mixture of the genuine and the chaff point set. To access secret data from the vault, an unordered set (Y) was used, which had to be almost equal to set X. If the difference between set X and set Y was very small, the genuine point set was discriminated from the vault by set Y. Finally, the polynomial (p) was perfectly reconstructed and the secret data was generated securely. In general, cryptographic systems require correct cryptographic keys in order to decrypt secret data. On the other hand, fuzzy vault scheme allows the fuzziness of the unordered sets (which act as secret keys), which means that biometric data can be used as unordered sets because biometric data contain the intra-variations of the same person. However, the fuzzy vault scheme also requires pre-aligned biometric templates that are properly aligned with the input biometric data [1].

Most previous works [1][6-9] that were based on Juels and Sudan's fuzzy vault scheme used fingerprint data. This is because fingerprints are reliable biometric features, and it is also easy to extract the proper features for the fuzzy vault. Clancy *et al*. [6] proposed a fingerprint-based fuzzy vault system. They used the location set of minutiae as an unordered set [3] and assumed that there were no great variations between the template and the query minutiae set. Uludag *et al*. [1] introduced a modified fuzzy vault system that did not require Reed-Solomon decoding and allowed manual alignment between the template and query fingerprint data. However, they also proposed a new method to automatically align the template and query fingerprint data with helper data [7]. Shibata *et al*. [8] proposed a minutiae-based fingerprint fuzzy vault system that used the clustering technique to automatically align fingerprint data. Yang *et al*. [9] proposed automatic fingerprint verification based on the fuzzy vault scheme. Freire-Santos *et al*. [10] proposed the implementation of the fuzzy vault system based on hand-written signatures. These researchers did not assume the pre-alignment of biometric data. Last, Feng *et al*. [11] proposed a fuzzy vault system that used face data.

3 Proposed Method

In this section, we present our method of generating unordered sets (a locking set and an unlocking set) and the implementation of our fuzzy vault system. Fig. 1 shows a block diagram of the proposed fuzzy vault system. Details explanations about Fig.1 are shown in section 3.2 and 3.3.

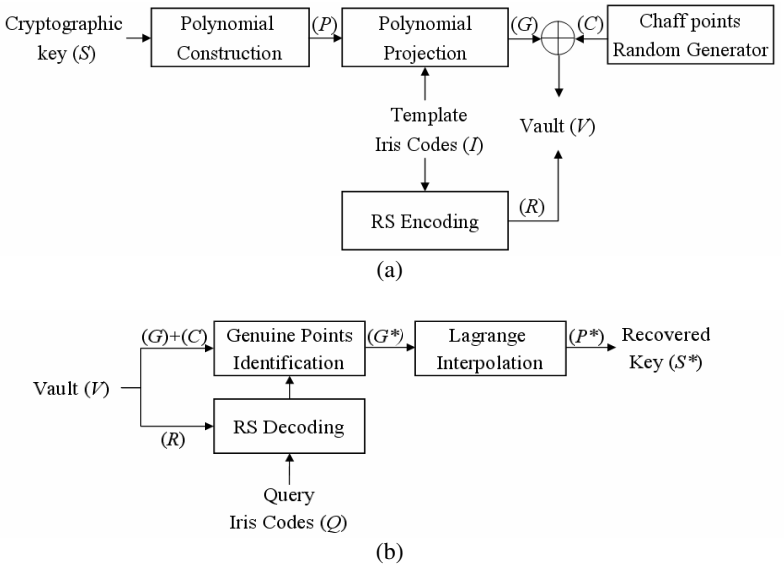
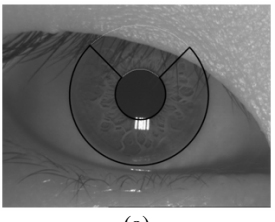


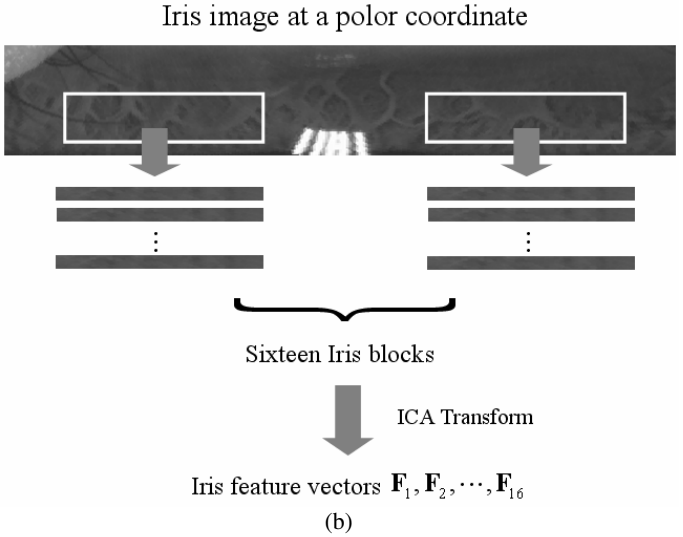
Fig. 1. Fuzzy vault system based on iris data: (a) locking the vault, (b) unlocking the vault

3.1 Feature Extraction

In order to generate an unordered set from iris data, we chose an algorithm [4-5] based on ICA. One reason is that ICA method is suitable for extracting multiple and local iris feature vectors from iris data. Another reason is that the performance of ICA method [4-5] is similar to the performance of Daugman's method [12]. Finally, instead of a global feature vector such as a 2048-bit iris code [12], we obtained multiple iris feature vectors from multiple iris image blocks using ICA method [4-5] because multiple input values are required for fuzzy vault system. The iris feature extraction process took place as follows: the first step was to localize the iris region in a captured eye image using a conventional circular edge detection method [4], as shown in Fig. 2(a). The second step was to transform the localized iris region into a polar coordinate to obtain the iris features invariant to translation and rotation. Then, we selected two iris regions which were not occluded by eyelids, eyelashes and specular reflections, as regions of interest for feature extraction, in the localized iris image at a polar coordinate, as shown in Fig. 2(b) [4].



(a)



(b)

Fig. 2. Iris feature extraction: (a) localized iris image, (b) dividing two selected iris regions into sixteen iris image sub-regions (blocks) in the localized iris image of a polar coordinate and extracting an iris feature vector from each iris image block

The last step was to divide each region into eight iris image blocks and to extract sixteen iris feature vectors from sixteen iris image blocks by using the ICA algorithm [4-5], as shown in Fig. 2(b). Here, sixteen extracted iris feature vectors were quantized to be 27-bit binary codes with a sign of ICA coefficients [4-5].

We used sixteen feature vectors (sixteen 27-bit binary codes) because a polynomial of 15 degrees was used in our fuzzy vault system (see section 3.2). However, we could not use a set of sixteen feature vectors as an unordered set because iris data generally contains intra-class variations and the elements of an unordered set must be elements of a finite field, often called the *Galois Field* (GF) [13] in accordance with Reed-Solomon codes (RS codes) [14-15], used for error correcting in fuzzy vault [3]. Therefore, we used a pattern clustering technique to reduce the variations between the iris templates and the input iris data, and proposed a method of generating a set of *Iris Codes* which were proper for an unordered set.

The set of *Iris Codes* was generated as follows: to produce a locking set [3], we obtained five eye images from the same user for clustering. Next, we made a cluster using five extracted iris feature vectors for each iris block. For clustering, the K-means algorithm [16] was used because the number of classes was known. After clustering, each user had a cluster for each iris block. Namely, one user obtained sixteen clusters for sixteen iris blocks. Last, to generate the *Iris Codes*, we assigned a random integer of the finite field $GF(2^8)$ to a prototypes of each cluster for each user. In our system, because we aimed at assigning 16 coefficients to 128 bits (for security level of fuzzy vault scheme to brute-force attacks, we do not use less than sixteen coefficients such as 8 or 4 coefficients), each coefficient had 8 bits, consequently. So, the *Galois Field* was defined as $GF(2^8)$ based on the principle of Reed-Solomon coding. Each user's sixteen *Iris Codes* produced this way were represented by 8-bit words and were the elements of the locking set. Also, an unlocking set consisted of the *Iris Codes* to be found by matching the input iris feature vectors with the prototypes of each cluster map.

3.2 Locking the Vault

The procedure of locking the vault required two input factors: a cryptographic key (S) as shown in Fig.1 and a given user's *Iris Codes* (I), which were extracted by the method discussed in section 3.1. In the implementation, S represented a 128-bit AES key [2] and I was composed of sixteen 8-bit words. Our fuzzy vault system is different from the fuzzy vault proposed by Juels and Sudan [3]. In our fuzzy vault system, the error correcting and interpolation procedures were separated, as shown in Fig. 1. The former procedure was performed by Reed-Solomon decoding [14-15] and the latter procedure was performed by the Lagrange interpolation technique [17].

A Detailed explanation of locking the vault is as follows. As shown in Fig. 1, S was used to construct the polynomial (P): the 128-bit key (S) was divided into no-overlapping 8-bit segments ($\{S_1, S_2, \ldots, S_{16}\}$) and sixteen segments were then used as coefficients of the polynomial (P). Hence, P was constructed with a degree $d = 15$ ($d = (B/l)-1$, where B represented the length of a key and l represented a bit-length of elements in a finite field): $P(x) = S_1 + S_2x + \ldots + S_{16}x^{15}$. Then, three sets ($G$, C, R) were generated, as shown in Fig. 1(a).

The first set was a genuine set G, which was formed by evaluating the polynomial $P(x)$ in terms of the *Iris Codes* ($I = \{c_1, c_2, \ldots, c_{16}\}$): $G = \{(c_1, P(c_1)), (c_2, P(c_2)), \ldots, (c_{16}, P(c_{16}))\}$. The set G was used to reconstruct the polynomial during unlocking. So, the size of the set G had to be sixteen (degree of the polynomial + 1) or more. The second set was the chaff point set C which had a significant role in protecting the genuine set G. The set C was generated randomly in the range of the finite field, with the constraint that their values on the x-axis could not overlap with the *Iris Codes* (I) and they could not be located on the polynomial (P). The last set was the redundancy set (R), obtained by RS encoding. The set R was used to correct errors at the time of unlocking the vault [15]. The set R was composed of 8-bit redundant symbols, and the number of symbols was determined as twice the number of errors to be corrected [15]. Finally, the three sets (G, C, R) were combined to create a vault (V) which was stored in a device such as a smartcard or a server.

3.3 Unlocking the Vault

To generate the cryptographic key (S), a user unlocked the vault with his (or her) queried *Iris Codes* (Q). Unlocking also needed two inputs: the vault (V) created at locking and the queried *Iris Codes* (Q), as shown in Fig. 1. The vault (V) was obtained from a storage device and the *Iris Codes* (Q) was generated using the procedure discussed in section 3.1. However, the *Iris Codes* (Q) contained some errors because of the variations of the iris images taken under different conditions in spite of using the clustering method. Therefore, at the beginning of unlocking the vault, we corrected the errors of the *Iris Codes* (Q) using the RS decoding algorithm [14-15].

At this time, set R (the redundancy set obtained by RS encoding, as shown in Fig. 1(a)) was also used for decoding. Then, if outcome (Q^*) of RS decoding was equal to a user's *Iris Codes* (I), the genuine set G (which was formed by evaluating the polynomial $P(x)$ by the *Iris Codes* (I), as described in section 3.2) was perfectly identified from $G+C$ (here, C represents the set of chaff points, as shown in Fig. 1(a)). Namely, the set G^* of Fig. 1(b) was equal to the set G. We then reconstructed the polynomial (P^*) with the set G^* by using Lagrange interpolation [17], a simple method that interpolates a polynomial with point pairs on the polynomial. When the point pairs were $(x_1, y_1), (x_2, y_2), \ldots, (x_{d+1}, y_{d+1})$, the corresponding polynomial was obtained as follows:

$$P(x) = \frac{f(x)}{(x-x_1)f'(x_1)}y_1 + \frac{f(x)}{(x-x_2)f'(x_2)}y_2 + \cdots + \frac{f(x)}{(x-x_{d+1})f'(x_{d+1})}y_{d+1}. \quad (1)$$

Here, $f(x) = (x- x_1)(x- x_2)\ldots(x- x_{d+1})$, $f'(x)$ represented a derivative of $f(x)$ and the degree of this polynomial was d (in our case, d was 15). In our paper, a set of point pairs was the genuine set G^* and a polynomial $P^*(x) = S^*_1 + S^*_2 x + \ldots + S^*_{16}x^{15}$ was reconstructed by Lagrange interpolation. Finally, all coefficients of the polynomial were concatenated as $S^*_1 S^*_2 \ldots S^*_{16}$ and the secret key S^* was recovered perfectly.

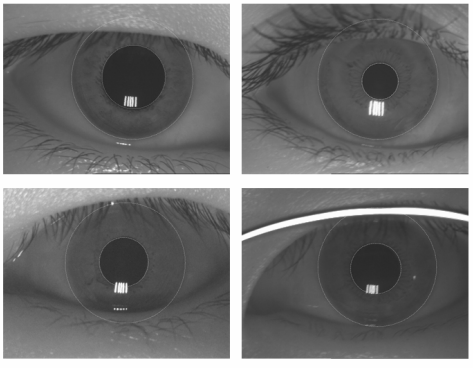


Fig. 3. Iris image examples from the BERC iris database (version 1)

4 Experimental Results

4.1 Iris Database

We used the BERC iris database (version 1) [18] to evaluate the proposed method. The BERC iris database (version 1) consists of 990 images: 10 images for each 99 individuals [4-5]. Fig. 3 shows some examples of the BERC iris database, captured at a resolution of 640×480 pixels and 8-bit gray information. These images were captured by our hand-made iris recognition camera, which contained a monochrome CCD sensor with a fixed focal lens, and an 850nm IR (Infra-Red) illuminator [4-5].

In our experiments, half of ten iris sample images per class were used for training and the remaining images were used for authentication. So, the number of authentic tests was 4,000 and the number of imposter tests was 16,000. We assumed that the difference between the iris templates and the input iris data was not great.

4.2 Experimental Results

In our experiments, the criteria of system performance were the False Rejection Rate (FRR) and the False Acceptance Rate (FAR). The FRR was defined as the error rate obtained when a legitimate user's cryptographic key was not generated when the user tried to obtain his or her key using his or her iris image and vault. The FAR was defined as the error rate obtained when a legitimate user's cryptographic key was generated when an illegitimate attacker attempted to steal a key using his or her iris image with a legitimate user's vault. The similarity between a queried iris feature vector and the prototype of each cluster was evaluated by Hamming distance.

In the conventional cryptographic system to which we tried to apply our fuzzy vault system, the FAR was more important than the FRR, because its main purpose is to make secret key for encryption system which can be used for banking service etc. So, a certain number of False Rejection cases can be accepted, such as when genuine users fail to input correct passwords. Therefore, in our experiments, we considered the minimum FRR value when the FAR was set to 0% as the optimal result.

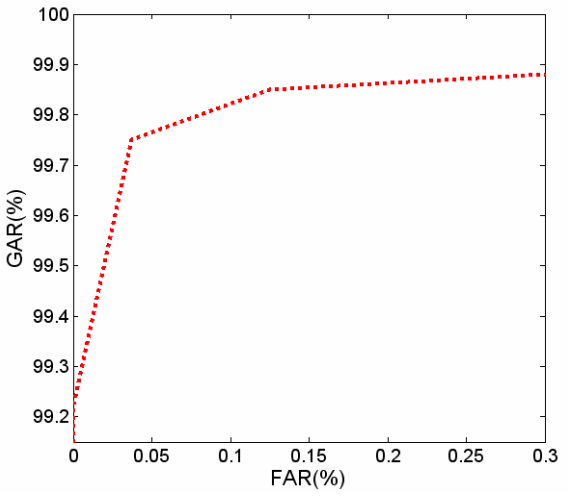


Fig. 4. ROC curves of the proposed fuzzy vault system

Fig. 4 shows the ROC curves. As shown in Fig. 4, when the FAR was set to 0%, the FRR was 0.775% (when the Genuine Acceptance Rate (GAR) was 99.225%), correcting eight errors among the sixteen queried *Iris Codes*.

5 Conclusions

The fuzzy vault system refers to a cryptographic method that encrypts and decrypts secret data with an unordered set. This cryptographic method can be used in combination with biometrics because it permits only a few variations of the unordered set by using error-correcting codes. In this paper, we proposed a new way of implementing the fuzzy vault system based on iris data.

In future work, we will evaluate our proposed method with regard to various input iris images (e.g., rotation, translation and blurring).

Acknowledgments. This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Biometrics Engineering Research Center (BERC) at Yonsei University.

References

1. Uludag, U., Pankanti, S., Jain, A.K.: Fuzzy Vault for Fingerprints. In: Proceedings of Audio- and Video-based Biometrics, December 4, 2006 DRAFT 31 Person Authentication, Rye Town, USA, July 2005, pp. 310–319 (2005)
2. Trappe, W., Washington, L.C.: Introduction to Cryptography with coding theory. Prentice-Hall, Upper Saddle River, NJ.
3. Juels, A., Sudan, M.: A fuzzy vault scheme. In: ACM Conference on Computer and Communications Security, CCS 2002. ACM, New York (2002)

4. Bae, K.H.: An Iris Feature Extraction Method using the Independent Component Analysis. The Graduate School of Yonsei University, Dept. of Electrical and Electronic Eng. (2002)
5. Bae, K.H., Noh, S.I., Park, Kim, J.H.: Iris Feature Extraction Using Independent Component Analysis. In: Kittler, J., Nixon, M.S. (eds.) AVBPA 2003. LNCS, vol. 2688, pp. 838–844. Springer, Heidelberg (2003)
6. Clancy, T., Lin, D., Kiyavash, N.: Secure Smartcard-Based Fingerprint Authentication. In: Proceedings of ACM SIGMM Workshop on Biometric Methods and Applications, Berkley, USA, November 2003, pp. 45–52. ACM, New York (2003)
7. Uludag, U., et al.: Securing Fingerprint Template: Fuzzy Vault With Helper Data. In: Proc. of CVPR Workshop on Privacy Research In Vision, New York, USA, June 2006, p. 163 (2006)
8. Shibata, Y., Nishigak, M.: A study on biometric key generation. Asian Biometrics Forum (2006)
9. Yang, S., et al.: Automatic Secure Fingerprint Verification System Based on Fuzzy Vault Scheme. In: Proc. of IEEE ICASSP, Philadelphia, USA, March 2005, vol. 5, pp. 609–612. IEEE, Los Alamitos (2005)
10. Freire-Santos, M., Fierrez-Aguilar, J., Ortega-Garcia, J.: Cryptographic Key Generation Using Handwritten Signature. In: Proceedings of Biometric Technologies for Human Identification III, Orlando, USA, April 2006, vol. 6202, pp. 225–231 (2006)
11. Feng, Y.C., Yuen, P.C.: Protecting Face Biometric Data on Smartcard with Reed-Solomon Code. In: Proceedings of CVPR Workshop on Privacy Research In Vision, New York, USA, June 2006, p. 29 (2006)
12. Daugman, J.: How iris recognition works. IEEE Trans. Circuits Syst. Video Techno. 14(1), 21–30 (2004)
13. Wicker, S.B.: Error Control Systems for Digital Communication and Storage. Prentice-Hall, Englewood Cliffs, NJ (1995)
14. Sylvester, J.: Reed-Solomon Codes (January 2001), Available from http://www.elektrobit.co.uk
15. Sklar, B.: Reed-Solomon Codes (accessed on 2007.2.10),.Available at, http://www.informit.com/content/images/art_sklar7_reed- solomon/elementLinks/art_sklar7_reed-solomon.pdf
16. Duda, R.O., Har, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, Chichester (2001)
17. Kreyszig, E.: Advanced Engineering Mathematics, 8th edn. Wiley, Chichester (1999)
18. (accessed on 2007.1. 2) http://berc.yonsei.ac.kr